

**Descripción del problema:** 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.

1.

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

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

<b>Nombre</b>	<b>R. 3. Hospitalizar a un cliente mascota</b>
<b>Resumen</b>	El usuario debe ingresar la información de un cliente mascota al sistema para poder ser utilizada dentro del programa y así hospitalizar una mascota, creandole una historia clínica
<b>Entradas</b>	
<ul style="list-style-type: none"> <li>-Fecha de ingreso del cliente mascota para la historia clinica</li> <li>- Síntomas presentados del cliente mascota para la historia clinica</li> <li>- El posible diagnóstico del cliente mascota para la historia clinica</li> <li>-Los medicamentos recetados del cliente mascota para la historia clinica</li> </ul>	
<b>Resultados</b>	
Hospitalizar a un cliente mascota en el sistema.	

<b>Nombre</b>	<b>R. 4. Registrar a un medicamento recetado</b>
<b>Resumen</b>	El usuario debe ingresar la información de un medicamento recetado al sistema para poder ser utilizada dentro del programa
<b>Entradas</b>	
<ul style="list-style-type: none"> <li>-El nombre</li> <li>-La dosis</li> <li>-El costo por dosis</li> <li>-La frecuencia con la cual se debe administrar al cliente mascota</li> </ul>	
<b>Resultados</b>	
Registrar la información de un medicamento recetado en el sistema.	

<b>Nombre</b>	<b>R. 5. Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales</b>
---------------	--

<b>Resumen</b>	El programa debe mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales	

<b>Nombre</b>	<b>R. 6. Mostrar los datos de contacto del dueño de un cliente mascota hospitalizado</b>
<b>Resumen</b>	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
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Mostrar los datos de contacto del dueño de un cliente mascota hospitalizado	

<b>Nombre</b>	<b>R. 7. Calcular el costo total de una hospitalización</b>
<b>Resumen</b>	El programa debe calcular el costo de una hospitalización por día, 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
<b>Entradas</b>	
<Ninguna>	
<b>Resultados</b>	

Costo total de la hospitalización
-----------------------------------

<b>Nombre</b>	<b>R. 8. Permitir dar de alta a un cliente mascota que haya estado hospitalizado</b>
<b>Resumen</b>	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
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Dar de alta a un cliente mascota que haya estado hospitalizado	

<b>Nombre</b>	<b>R. 9. Mostrar datos de hospitalización de un cliente mascota</b>
<b>Resumen</b>	El programa debe seguido al dar de alta, mostrar los datos de hospitalización de un cliente mascota
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Mostrar datos de hospitalización de un cliente mascota	

<b>Nombre</b>	<b>R. 10. Poner disponible un minicuarto</b>
<b>Resumen</b>	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
<b>Entradas</b>	
<Ninguno>	

Resultados	
Disponible el minicuarto que estaba siendo ocupado	

<b>Nombre</b>	<b>R. 11. Mostrar ingresos por concepto de hospitalizaciones</b>
<b>Resumen</b>	El programa debe sumar todos los cobros hechos por concepto de hospitalizaciones para poder ser mostrado posteriormente
Entradas	
<Ninguno>	
Resultados	
Mostrar ingresos por concepto de hospitalizaciones	

<b>Nombre</b>	<b>R. 12. Mostrar el número del mini cuarto que ocupa un cliente mascota hospitalizado</b>
<b>Resumen</b>	El programa debe mostrar el número del mini cuarto que ocupa un cliente mascota hospitalizado según en su nombre.
Entradas	
<Ninguno>	
Resultados	
Mostrar el número del mini cuarto que ocupa un cliente mascota hospitalizado	

<b>Nombre</b>	<b>R. 13. Mostrar el historial de historias medicas de una mascota</b>
<b>Resumen</b>	El programa debe mostrar el historial de historias medicas de una mascota
Entradas	
-ID del dueño -Nombre del dueño	

-Nombre de la mascota
<b>Resultados</b>
Mostrar el historial de historias medicas de una mascota

<b>Nombre</b>	<b>R. 14. Calcular los ingresos por servicios</b>
<b>Resumen</b>	El programa debe poder calcular el ingreso total por concepto de servicios prestados
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Ingreso total por concepto de servicios prestados	

<b>Nombre</b>	<b>R. 15. Calcular los ingresos totales de la veterinaria</b>
<b>Resumen</b>	El programa debe poder calcular el ingreso total por concepto de servicios prestados y hospitalizaciones
<b>Entradas</b>	
<Ninguno>	
<b>Resultados</b>	
Ingreso total por concepto de servicios prestados y hospitalizaciones	

<b>Nombre</b>	<b>R. 16. Agregar al sistema nuevos servicios vendidos por la veterinaria</b>
<b>Resumen</b>	El usuario debe ingresar la información de un servicio prestado al sistema para poder ser utilizada dentro del programa
<b>Entradas</b>	

-Tipo de servicio -ID del dueño de la mascota -Nombre de la mascota -Fecha de la prestación del servicio
<b>Resultados</b>
Registrar un servicio prestado en el sistema.

<b>Nombre</b>	<b>R. 17. Mostrar promedio de ingresos de la veterinaria en una semana</b>
<b>Resumen</b>	El programa debe poder calcular el promedio de ingresos de la veterinaria en una semana dada una fecha inicial
<b>Entradas</b>	
-Fecha inicial de una semana	
<b>Resultados</b>	
Promedio de ingresos de la veterinaria en una semana dada una fecha inicial.	

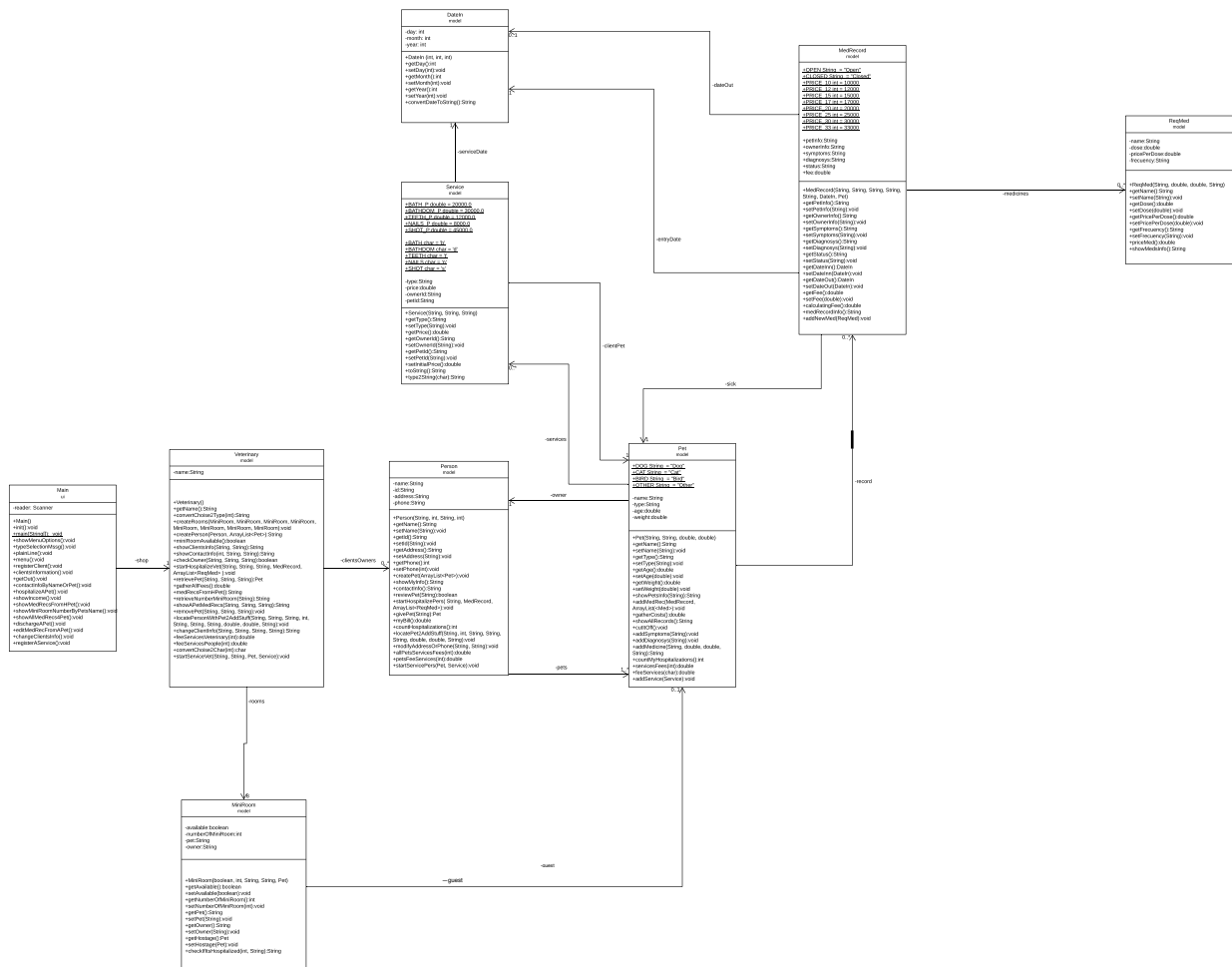
<b>Nombre</b>	<b>R. 18. Mostrar reporte de servicios prestados dada una fecha inicial y una fecha final.</b>
<b>Resumen</b>	El usuario debe ingresar la información de un cliente humano al sistema para poder ser utilizada dentro del programa
<b>Entradas</b>	
-Fecha inicial -Fecha final	
<b>Resultados</b>	
Reporte de servicios prestados dada una fecha inicial y una fecha final.	

## REQUERIMIENTOS NO FUNCIONALES

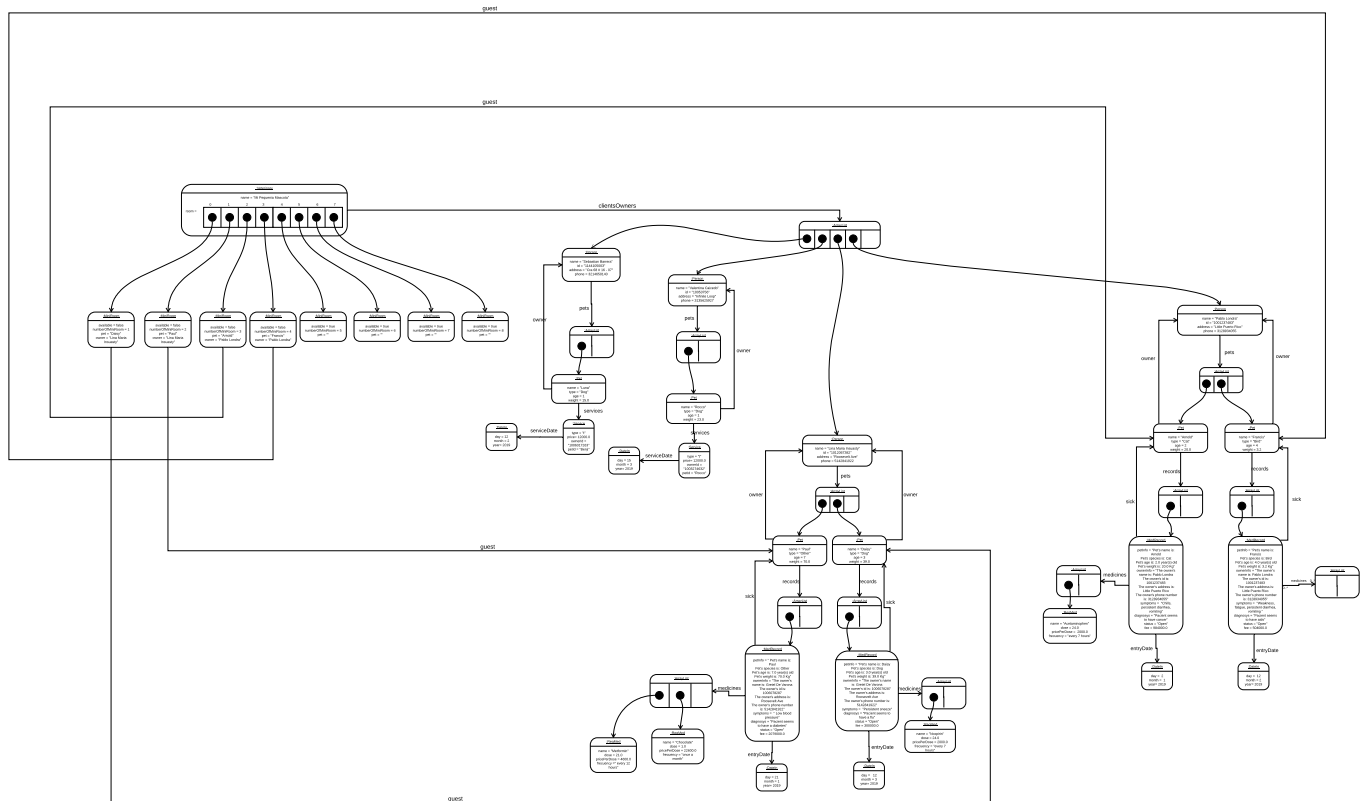
<b>Nombre</b>	<b>RNF 1. 8 mini cuartos</b>
<b>Resumen</b>	La veterinaria cuenta únicamente con 8 mini cuartos.

<b>Nombre</b>	<b>RNF 2. Una mascota por mini cuarto</b>
<b>Resumen</b>	Cada mini cuarto solo puede alojar a una mascota.

<b>Nombre</b>	<b>RNF 3. Historial medico permanente</b>
<b>Resumen</b>	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







## TRAZABILIDAD

REQUERIMIENTO FUNCIONAL	CÓDIGO	
	CLASE	MÉTODO
R. 1. Registrar a un cliente persona	Main Veterinary Person	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +registerANewClient():void +convertChoise2Type(int):String +createPerson(Person, ArrayList<Pet>):void +createPet(ArrayList<Pet>):void
R. 2. Registrar a un cliente mascota	Main Veterinary Person	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +registerANewClient():void +convertChoise2Type(int):String +createPerson(Person, ArrayList<Pet>):void +createPet(ArrayList<Pet>):void

R. 3. Hospitalizar a un cliente mascota	Main Veterinary MiniRoom Person Pet MedRecord DateIn ReqMed	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +hospitalizeAPet():void +miniRoomAvailable():boolean +getAvailable():boolean +checkOwner(String, String, String)boolean +showClientsInfo(String, String):String +getName():String +getId():String +showMyinfo():String +showPetsinfo():String +retrievePet(String, String, String):Pet +getName():String +getId():String +givePet(String):Pet +getName():String +startHospitalizeVet(String, String, String, MedRecord, ArrayList<ReqMed>):void +getName():String +getId():String +startHospitalizePers(String, MedRecord, ArrayList<ReqMed>):void +getName():String +addMedRec(MedRecord, ArrayList<ReqMed>):void +setPetInfo(String):void +showPetsinfo():String +setOwnerInfo(String):void +contactInfo():String +addMeds(ArrayList<ReqMed>):void +calculatingFee():double +getDay():int +getMonth():int +getYear():int +getType():String +getWeight():double +priceMed():double
R. 4. Registrar a un medicamento recetado	Main Veterinary MiniRoom Person Pet MedRecord	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +hospitalizeAPet():void +miniRoomAvailable():boolean +getAvailable():boolean +checkOwner(String, String, String)boolean +showClientsInfo(String, String):String +getName():String +getId():String +showMyinfo():String +showPetsinfo():String +retrievePet(String, String, String):Pet

Página 11 de 16 - Algoritmos y Programación I - Universidad Icesi

R. 5. Mostrar la información de todas las historias clínicas de los clientes mascotas hospitalizados actuales	Main Veterinary MiniRoom Pet MedRecord ReqMed DateIn	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showMedRecsFromHPet():void +medRecsFromHPet():String +getAvailable():boolean +getHostage():Pet +showAllRecords():String +medRecordInfo():String +convertDateToString():String +showMedsinfo():String
R. 6. Mostrar los datos de contacto del dueño de un cliente mascota hospitalizado	Main Veterinary MiniRoom Person Pet	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +contactInfoByNameOrPet():void +typeSelectionMssg():void +plainLine():void +showContactInfo(int, String, String):String +checkIfItsHospitalized(int, String):String +getName():String +getId():String +contactInfo():String +checkIfItsHospitalized(int, String):String +getName():String +getId():String +getOwner():String +getName():String +reviewPet(String):boolean +getName():String +contactInfo():String
R. 7. Calcular el costo total de una hospitalización	MedRec ReqMed DateIn	+calculatingFee():double +getDay():int +getMonth():int +getYear():int +getType():String +getWeight():double +priceMed():double
R. 8. Permitir dar de alta a un cliente mascota que haya estado hospitalizado	Main Veterinary MiniRoom Person Pet DateIn MedRec ReqMed	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +dischargeAPet():void +checkOwner(String, String, String):boolean +showClientsInfo(String, String):String +getName():String +getId():String +showMyinfo():String +showPetsinfo():String +retrieveNumberMiniRoom(String):int

		+getPet():String +getNumberOfMiniRoom():int +showAPetMedRecs(String, String, String):String +getName():String +getId():String +givePet(String):Pet +showAllRecords():String +medRecordInfo():String +convertDateToString():String +showMedsinfo():String +removePet(String, String, String):void +getName():String +getId():String +givePet(String):Pet +getName():String +cutItOff():void +setStatus(String):void +setDateOut(DateIn):void +getPet():String +getOwner():String +setAvailable(boolean):void +setPet(String):void +setOwner(String):void +setHostage(Pet):void
R. 9. Mostrar datos de hospitalización de un cliente mascota	Main Veterinary Person Pet DateIn ReqMed MedRec	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showAllMedRecs4Pet():void +checkOwner(String, String, String):boolean +showClientsInfo(String, String):String +getName():String +getId():String +showMyinfo():String +showPetsinfo():String +showAPetMedRecs(String, String, String):String +getName():String +getId():String +givePet(String):Pet +getName():String +showAllRecords():String +medRecordInfo():String +convertDateToString():String +showMedsinfo():String
R. 10. Poner disponible un minicuarto	Veterinary MiniRoom	+removePet(String, String, String):void +getName():String

	Person Pet MedRecord	+getId():String +givePet(String):Pet +getName():String +cutItOff():void +setStatus(String):void +setDateOut(DateIn):void +getPet():String +getOwner():String +setAvailable(boolean):void +setPet(String):void +setOwner(String):void +setHostage(Pet):void
R. 11. Mostrar ingresos por concepto de hospitalizaciones	Main Veterinary Person Pet MedRec	+showIncome():void +gatherAllFees():double +myBill():double +gatherCosts():double +getFee():double
R. 12. Mostrar el número del mini cuarto que ocupa un cliente mascota hospitalizado	Main Veterinary MiniRoom	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showMiniRoomNumberByPetsName():void +retrieveNumberMiniRoom(String):int +getPet():String +getNumberOfMiniRoom():int
R. 13. Mostrar el historial de historias medicas de una mascota	Main Veterinary Person Pet MedRec ReqMed DateIn	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +checkOwner(String, String, String):boolean +showClientsInfo(String, String):String +getName():String +getId():String +showMyinfo():String +showPetsinfo():String +showAPetMedRecs(String, String, String):String +getName():String +getId():String +givePet(String):Pet +showAllRecords():String +medRecordInfo():String +convertDateToString():String +showMedsinfo():String

R. 14. Calcular los ingresos por servicios	Main Veterinary Person Pet Service MedRec	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showIncome():void +gatherAllHospitalizationFees():double +feeServicesVeterinary(int):double +feeServicesPeople(int):double +petsFeeServices(int):double +servicesFees(int):double +feeServices(char):double +getType():char +getPrice():double
R. 15. Calcular los ingresos totales de la veterinaria	Main Veterinary Person Pet Service MedRec	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showIncome():void +totalIncome():double +gatherAllHospitalizationFees():double +feeServicesVeterinary(int):double +feeServicesPeople(int):double +petsFeeServices(int):double +servicesFees(int):double +feeServices(char):double +getType():char +getPrice():double +myBill():double +gatherCosts():double +getFee():double
R. 16. Agregar al sistema nuevos servicios vendidos por la veterinaria	Main Veterinary Person Pet Service	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +registerAService():void +convertChoise2Char(int):char +retrievePet(String, String, String):Pet +findPerson(String, String):Person +getName():String +givePet(String):Pet +getName():String +startServiceVet(String, String, Pet, Service):void +startServicePers(Pet, Service):void +addService(Service):void
R. 17. Mostrar promedio de ingresos de la veterinaria en una semana	Main Veterinary Person Pet Service DateIn	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showAverageIncomeByWeek():void +weeklyAverageIncome(DateIn):double +petsServicesWeeklyIncome(DateIn):double

		+weeklyIncome(DateIn):double +calculateDayOfTheYear():int +getDateJob():DateIn +getYear():int +dateJobDayOfYear():int +getPrice():double +petsServicesWeeklyIncomeCounter(DateIn):int +weeklyIncomeCounter(DateIn):double +calculateDayOfTheYear():int +getDateJob():DateIn +getYear():int +dateJobDayOfYear():int
R. 18. Mostrar reporte de servicios prestados dada una fecha inicial y una fecha final	Main Veterinary Service DateIn	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void