

Laboratorio Unidad 3

En la unidad 3 de nuestro curso hemos trabajado sobre **estructuras contenedoras de tamaño fijo y variable**, de manera que aprendimos **cómo modelar dichas estructuras en un diagrama de clases, e implementar soluciones haciendo uso de ellas**. Además, aprendimos **como utilizar instrucciones iterativas para manipular estructuras contenedoras o para resolver problemas que requieren la repetición de un conjunto de instrucciones**. El presente laboratorio les presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos en esta unidad y verificar de esta manera el cumplimiento de los objetivos que han sido planteados para la unidad 3 descritos en el [programa del curso](#). Para llevar a cabo este ejercicio es necesario realizar las actividades listadas a continuación:

Actividades

Lleve a cabo las siguientes actividades de cada una de las etapas de desarrollo de software:

1. Análisis del problema (Definición del problema, identificación de entidades, sus características y relaciones) y especificación de Requerimientos Funcionales
2. Diagrama de Clases Completo (incluye el Modelo y el Main en la interfaz). El modelo debe ser elaborado digitalmente, pero NO generado automáticamente (por ejemplo, no es válido entregar modelos generados por Object Aid o ninguna otra herramienta).
3. **Diagrama de Objetos de la situación inicial** de su software.
4. Trazabilidad del Análisis al Diseño. Una tabla a **dos columnas** en la que se relaciona **cada requerimiento con el método o métodos que permiten satisfacer dicho requerimiento**.
5. Implementación en Java. Incluya en la implementación, los comentarios descriptivos sobre los atributos y métodos de cada clase. Recuerde que todos los artefactos generados de fase de diseño e implementación deben ser en inglés.
6. Usar GitHub como repositorio de código fuente utilizando la estructura de carpetas aprendida en clase. Recuerde que se debe evidenciar su avance a través de los días en el laboratorio.

Recuerde que puede encontrar la Rúbrica laboratorio en el siguiente [enlace](#).

Se espera que usted haya realizado la actividad 1 para el día viernes 8 de marzo, se realizará la revisión de esta actividad.

Para llevar a cabo la actividad 1 recuerde que:

1. Los sustantivos describen posibles candidatos a entidades.
2. Los verbos asociados a posibles entidades sugiere los requerimientos funcionales.
3. Las relaciones usualmente están dadas cuando encuentre frases como: “se relaciona con” o “tiene”.

Usted debe subrayar con diferentes colores las entidades, sus características, sus relaciones y los requerimientos funcionales identificados.

Nota: Usted debe entregar un archivo en formato pdf con toda la documentación (análisis, diseño y tabla de trazabilidad) y la URL de su repositorio git Hub donde se deben encontrar los archivos de codificación en sus respectivos paquetes.

Tenga en cuenta que su repositorio gitHub debe presentar una estructura base como por ejemplo:

```
Veterinary/  
    src/      bin/      docs/  
    model/    ui/
```

En la carpeta src (source code) contendrá las carpetas model y ui, donde usted deberá almacenar sus clases .Java

Su código debería compilar de acuerdo con lo explicado en la diapositiva 13 de esta presentación:

<http://tinyurl.com/y3bd9bg2>

Enunciado

Gris clases

Amarillo requerimientos

Verde atributos

Azul relaciones

Azul mar Constantes

En la veterinaria *“Mi pequeña mascota”* han identificado la necesidad de una aplicación de software para ayudar con el funcionamiento de su negocio. La veterinaria ha trabajado en la ciudad de Cali durante más de 10 años y presta diversos productos y servicios a sus clientes. Debido a ser un negocio pequeño han decidido partir el proyecto del desarrollo de la aplicación en dos fases y le han solicitado a usted ayuda con la fase 1 del proyecto.

La primera fase del proyecto consta en modelar el funcionamiento de los procesos básicos de la veterinaria. La veterinaria desea poder registrar a sus clientes humanos y sus mascotas. De cada persona se desea conocer el nombre, la identificación, la dirección, y el teléfono de contacto. Una persona puede tener 1 o más mascotas, y de cada mascota se desea conocer el nombre, el tipo de animal (Por ahora se especifican perro, gato, ave y otros), la edad y el peso.

Entre los servicios que ofrece la veterinaria se encuentra la hospitalización, para lo cual tiene habilitados unos doomini cuartos para las mascotas, específicamente tiene 8. Para hospitalizar una mascota se requiere saber si hay disponibilidad de un mini cuarto, en caso de haber disponibilidad, se debe crear la historia clínica de la mascota, con los datos iniciales correspondientes a la mascota (nombre de la mascota, el peso del animal, el tipo de animal, la edad del animal y el peso del animal), su dueño (nombre, la identificación, la dirección, y el teléfono de contacto), un estado (ABIERTA o CERRADA, las historias

clínicas se crean con un estado ABIERTA y se cambian a estado CERRADA cuando se da de alta al animalito), la fecha de ingreso, los síntomas presentados, el posible diagnóstico y los medicamentos recetados. De cada medicamento recetado se desea conocer el nombre, la dosis, el costo por dosis y la frecuencia con la cual se debe administrar al animalito.

Como parte de las funciones requeridas en la primera fase del proyecto se requiere poder realizar un informe de las historias clínicas de los pacientes hospitalizados en el momento de la consulta del reporte. Consultar los datos de contacto del dueño de un animalito hospitalizado a partir del nombre de su dueño o del nombre del animalito. Además, se desea calcular el costo de una hospitalización, teniendo en cuenta que cada día de hospitalización tiene un costo, el cual se calcula de acuerdo a la siguiente tabla.

Peso (kg)	Tipo de Animal	Valor por cada día
1 - 3	Gato	\$10.000
3.1 - 10		\$12.000
10.1 - 20		\$15.000
Más		\$20.000
1 - 3	Perro	\$15.000
3.1 - 10		\$17.000
10.1 - 20		\$20.000
1 - 3		\$25.000
1 - 3	Ave	\$10.000
3.1 - 10		\$12.000
10.1 - 20		\$20.000
1 - 3		\$25.000
1 - 3	Otros	\$10.000
3.1 - 10		\$17.000
10.1 - 20		\$30.000

1 - 3		\$30.000
-------	--	----------

Al costo de la hospitalización diaria se le debe sumar el costo de los medicamentos, de acuerdo a la cantidad de dosis del medicamento que se le han aplicado al animalito.

La aplicación debe permitir dar de alta a un animalito que haya estado hospitalizado, una vez se da de alta a un paciente se elimina la relación de su historia clínica con el mini cuarto y se debe guardar en un historial de historias clínicas, por lo cual al dar de alta un paciente la veterinaria desea entregar un informe de los datos de la hospitalización del paciente (Salidas) y poner disponible el mini cuarto. Para la veterinaria también es importante conocer cuánto han sido sus ingresos por concepto de hospitalizaciones y permitir saber el número del mini cuarto que ocupa una mascota basado en su nombre.

Finalmente, la veterinaria le ha solicitado poder consultar en el historial de historias clínicas si una mascota ya ha tenido una hospitalización anterior cada vez que se va a crear una nueva historia clínica, en caso de que exista la historia clínica anterior debe anexarse en la nueva historia clínica.

Cuando yo creo una historia clínica debo de preguntar si ya el animal tiene historia clínica. Si ya existe se anexa la nueva historia.

Repositorio git Hub:

<https://github.com/CamiloEscobar33/VeterinaryCali.git>

Requerimientos:

Nombre	RF1: Permite registrar dueños de mascotas.
Resumen	Permite registrar a los dueños de la mascota.
Entrada	Nombre cliente, id, dirección, teléfono de contacto.
Salida	Mostrar en pantalla el cliente fue registrado correctamente.

Nombre	RF2: Añadir una mascota a la veterinaria .
Resumen	Lograr meter una mascota en la veterinaria.
Entrada	Nombre de la mascota, tipo de animal, edad y peso.
Salida	Mostrar en pantalla que la mascota fue añadida correctamente.

Nombre	RF3: Crear historia clínica de una mascota.
Resumen	Crear una historia clínica de una mascota colocando todo su respectiva información, en caso de que exista la historia clínica debe anexarse en la nueva historia clínica.
Entrada	Datos correspondientes a la mascota, su dueño, el estado, fecha de ingreso, los síntomas presentados, el posible diagnóstico y los medicamentos recetados.
Salida	Imprimir en pantalla “Se le ha creado las historia clínica a la mascota” o “ Se ha anexado correctamente a la historia clínica a la mascota”.

Nombre	RF4: Determina la disponibilidad de los mini Cuartos.
Resumen	Permite determinar si hay disponibilidad de un mini cuarto.
Entrada	---
Salida	Mostrar los mini Cuartos que están disponibles.

Nombre	RF5: Informe de historias clínicas de pacientes hospitalizados.
Resumen	Permite mostrar un informe de las historias clínicas de los pacientes que han sido hospitalizados.
Entrada	
Salida	Mostrar el informe.

Nombre	RF6: Permite consultar los datos de contacto del dueño de un animalito hospitalizado.
Resumen	Consultar los datos del dueño al cual tiene una mascota hospitalizada.
Entrada	Id del dueño.
Salida	Datos de contacto del dueño de la mascota.

Nombre	RF7: Calcular el costo de una hospitalización.
Resumen	Teniendo en cuenta que cada día de hospitalización tiene un costo y con esto poder calcular el costo de la hospitalización.
Entrada	Id del dueño, nombre del animal hospitalizado, fecha de entrada y fecha salida.
Salida	Costo de hospitalización del animal.

Nombre	RF8: Calcular el costo de los medicamentos
Resumen	Permite calcular el costo de los medicamentos teniendo en cuenta la cantidad de dosis del medicamento que se le han aplicado al animalito.
Entrada	id del dueño, nombre del animal.
Salida	Mostrar en pantalla el costo del medicamento.

Nombre	RF9: Permitir dar alta a un animalito que está hospitalizado.
Resumen	Permite dar de alta a un animal que ha estado hospitalizado y mostrar un informe de datos del animal hospitalizado, y mostrar un mensaje que se le ha guardado su caso en un historial de historias clínicas.
Entrada	Id del dueño, nombre del animal.
Salida	Mostrar en pantalla “(nombre del animal) ha sido de alta exitosamente” y un informe de datos de la hospitalización del paciente.

Nombre	RF10: Determina cuánto han sido sus ingresos por concepto de hospitalizaciones
Resumen	Permitir saber cuál es la cantidad monetaria total de las hospitalizaciones.
Entrada	.-
Salida	Ingreso totales de hospitalizaciones.

Nombre	RF11: Permitir saber el número del mini cuarto que ocupa una mascota basado en su nombre.
Resumen	Permite con su nombre determinar en qué Mini cuarto esta la mascota.
Entrada	Nombre animal.
Salida	Numero de mini cuartos que esta la mascota.

Nombre	RF12: Permite consultar en el historial de historias clínicas la cantidad de hospitalizaciones que ha tenido un animal.
Resumen	Determinar la cantidad de hospitalizaciones que ha tenido una mascota.
Entrada	Id del dueño y nombre de la mascota.
Salida	Numero de hospitalizaciones de la mascota.

Diagrama de clases:

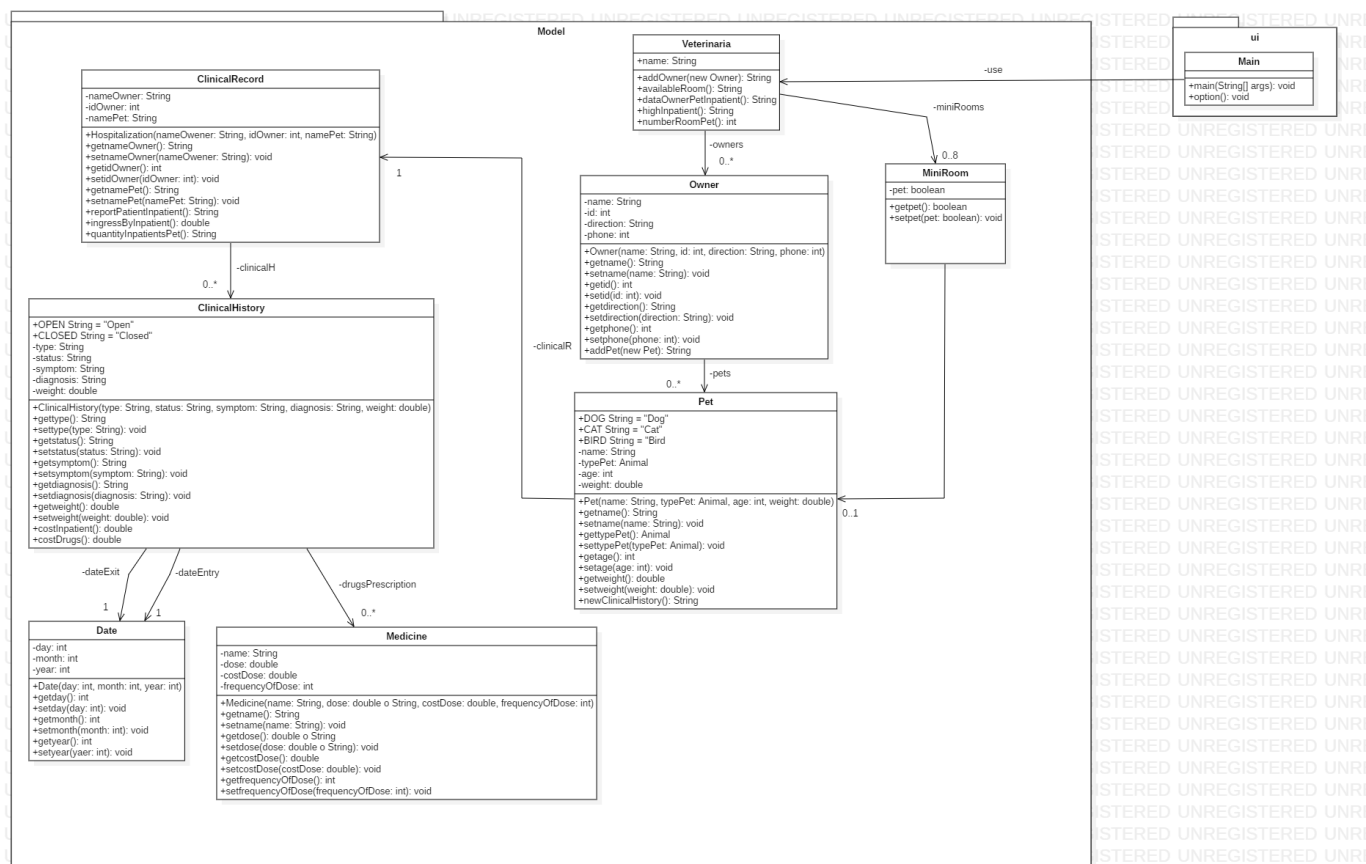
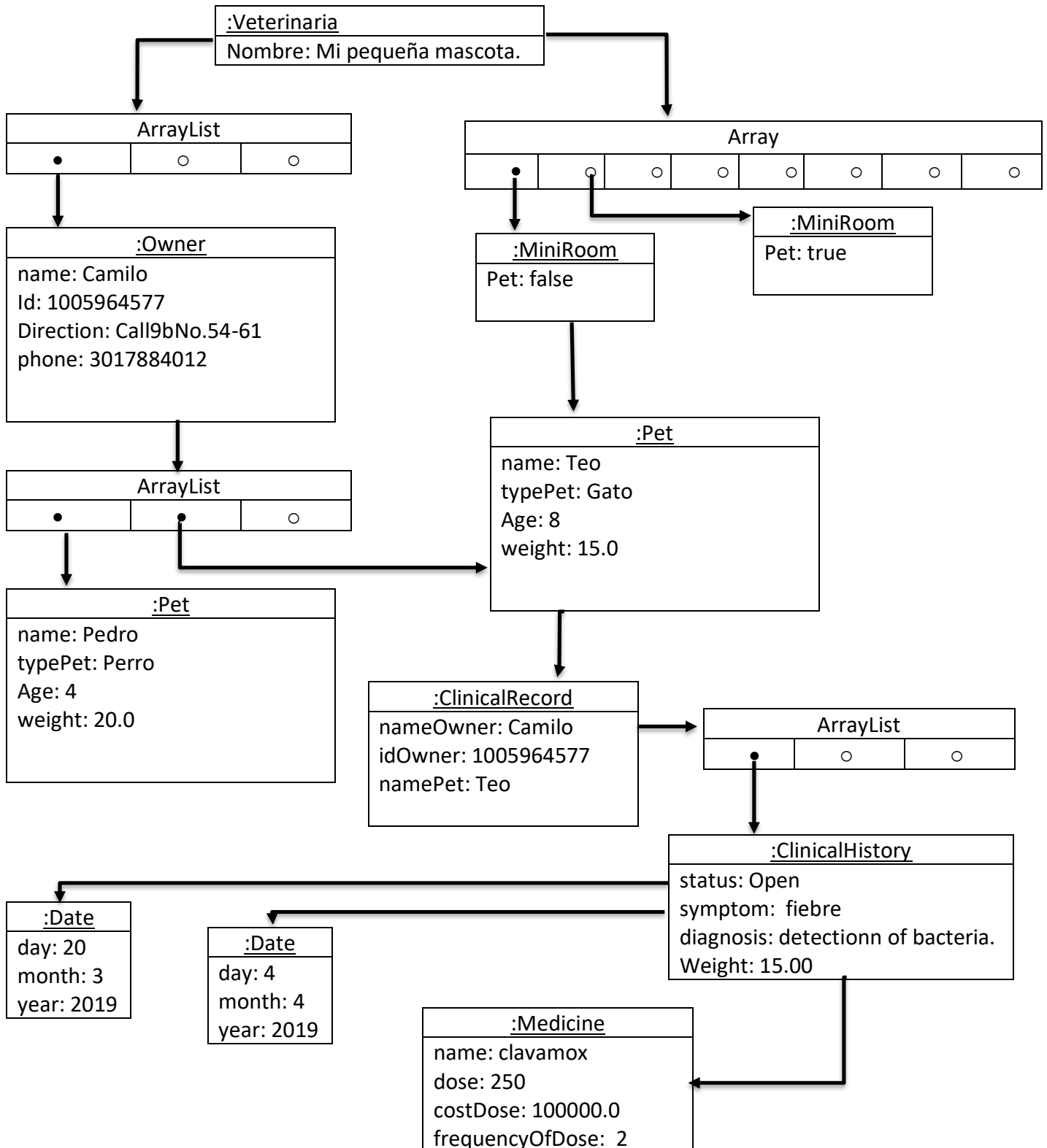


Diagrama de objetos:

(En StarUML no me salda nada de cómo hacer las figuras de los arreglos por eso lo hice aquí y con ese tipo de flecha).



Trazabilidad del Análisis al Diseño:

Requerimiento	Metodos
RF1: Permite registrar dueños de mascotas.(addOwner)*	
RF2: Añadir una mascota a la veterinaria. (addPet)*	
RF3: Crear historia clínica de una mascota. (newClinicalHistory)*	
RF4: Determina la disponibilidad de los mini Cuartos. (availableRoom)*	
RF5: Informe de historias clínicas de pacientes hospitalizados. (reportPatientInpatient)*	
RF6: Permite consultar los datos de contacto del dueño de un animalito hospitalizado. (dataOwnerPetInpatient)*	
RF7: Calcular el costo de una hospitalización. (costInpatient)*	
RF8: Calcular el costo de los medicamentos. (costDrugs)*	
RF9: Permitir dar alta a un animalito que está hospitalizado. (highInpatient)*	
RF10: Determina cuánto han sido sus ingresos por concepto de hospitalizaciones (ingressByInpatient)*	
RF11: Permitir saber el número del mini cuarto que ocupa una mascota basado en su nombre. (numberRoomPet)*	
RF12: Permite consultar en el historial de historias clínicas la cantidad de hospitalizaciones que ha tenido un animal. (quantityInpatientsPet)*	

