

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>

ANDREA NUÑEZ RODRIGUEZ

CLASES

REQUERIMIENTOS

ATRIBUTOS

RELACIONES

CONSTANTES

Enunciado

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 mini 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 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.

LABORATORIO 3

LISTA DE POSIBLES REQUERIMIENTOS FUNCIONALES

Nombre	R.# 1. Registrar a los clientes humanos y a sus respectivas mascotas
Resumen	El programa debe permitir que el usuario registre los datos de sus clientes humanos y a sus mascotas
Entradas: <ul style="list-style-type: none"> Datos del dueño Datos de la mascota 	
Resultados: Los datos del cliente humano y su mascota han sido registrados	

Nombre	R.# 2. Conocer la disponibilidad de un minicuarto
Resumen	El programa permite que el usuario saber la disponibilidad del minicuarto para seguidamente poder hospitalizar a una mascota.
Entradas:	
Resultados: Se ha verificado la disponibilidad de un minicuarto.	

Nombre	R.#3. Crear historia clínica de las mascotas
Resumen	El programa permite que el usuario ingrese la información de la mascota para poder hacer futuras operaciones con ella.
Entradas: <ul style="list-style-type: none"> Datos de la mascota Datos del dueño Fecha de ingreso Síntomas de la mascota Posible diagnostico Medicamentos 	
Resultados: La historia clínica de la mascota ha sido creada.	

Nombre	R.# 4. Dar de alta a la mascota
Resumen	El programa permite que el usuario dé de alta a la mascota en cuanto esta se sienta mejor.
Entradas:	
<ul style="list-style-type: none"> Nombre de la mascota a dar de alta. 	
Resultados: La relación de la mascota con el minicuarto que se le había asignado ha sido eliminada.	

Nombre	R.# 5. Realizar informe de las historias clínicas de las mascotas.
Resumen	El programa realiza un informe de las historias clínicas de las mascotas hospitalizadas en el momento de las consultas del reporte.
Entradas:	
Resultados: El informe de la historia clínica de la mascota ha sido creado.	

Nombre	R.# 6. Calcular el costo de una hospitalización
Resumen	El programa debe calcular el costo de la hospitalización de un animal dependiendo del día en que se esté.
Entradas:	
<ul style="list-style-type: none"> Peso del animal Tipo de animal Valor por día 	
Resultados: Costo de la hospitalización	

Nombre	R.# 7. Conocer los ingresos por hospitalizaciones.
Resumen	El programa permite que el usuario conozca la información de los ambientes de los canguros y de los dragones.
Entradas:	
Resultados: Los ingresos por hospitalizaciones han sido cargados.	

Nombre	R.# 8. Saber el número del cuarto en el cual está ubicada cada mascota
Resumen	El programa permite que el usuario conozca la ubicación del cuarto de la mascota ingresando su nombre.
Entradas:	
<ul style="list-style-type: none"> Nombre de la mascota 	
Resultados: El nombre del cuarto en el cual está la mascota.	

Nombre	R.# 9. Consultar el historial de una mascota.
Resumen	El programa permite que el usuario consulte el historial de una mascota teniendo en cuenta las hospitalizaciones que esta haya tenido, cada vez que vaya a crear una nueva historia clínica y en caso de que ya exista, debe anexarla a la anterior.
Entradas:	
Resultados: La historia clínica ha sido anexada.	

TRAZABILIDAD

REQUERIMIENTO	MÉTODOS	CLASES
R.# 1. Registrar a los clientes humanos y a sus respectivas mascotas	+setPetClient.add(ArrayList<Pet> petClient):void +setOwner.add(Client owner):void	Pet Client
R.# 2. Conocer la disponibilidad de un minicuarto	+ShowRoomsAvaliables():String	Veterinary
R.#3. Crear historia clínica de las mascotas	+setThePetRecord(Pet thePetRecord):void +setThePet(MedicalRecord thePet):void +setTheClient(Client theClient):void	MedicalRecord MedicalHistory
R.# 4. Dar de alta a la mascota	+setStatus(String status):void +setPetRoom(Pet petRoom):void	MedicalRecord Room Main

R.# 5. Realizar informe de las historias clínicas de las mascotas.	+ showInformationHistorial():String	Veterinary
R.# 6. Calcular el costo de una hospitalización	+calculateCostOfHospitalization():String <u>+getTotalDays()</u> <u>+priceMedicineTotal</u>	Veterinary MedicalRecord Medicine
R.# 7. Conocer los ingresos por hospitalizaciones.	+ calculateEarnings()	Veterinary
R.# 8. Saber el número del cuarto en el cual está ubicada cada mascota	+showInformation():String	Veterinary
R.# 9. Consultar el historial de una mascota.	+ showInformationHistorial()	Veterinary