

LABORATORIO 3

GONZALO DE VARONA

A00358687

ALGORTIMOS Y PROGRAMACIÓN 1

UNIVERSIDAD ICESI

MARZO 27 DE 2019

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

ENTIDADES Y SUS CARACTERÍSTICAS :

RELACIONES:

RF:

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.

DESARROLLO

1.

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, creandole una historia clínica
Entradas	
-Fecha de ingreso del cliente mascota para la historia clinica - Síntomas presentados del cliente mascota para la historia clinica - El posible diagnóstico del cliente mascota para la historia clinica -Los medicamentos recetados del cliente mascota para la historia clinica	
Resultados	
Hospitalizar a un cliente mascota en el sistema.	

Nombre	R. 4. Registrar a un medicamento recetado
Resumen	El usuario debe ingresar la información de un medicamento recetado al sistema para poder ser utilizada dentro del programa

Entradas
<ul style="list-style-type: none"> -El nombre -La dosis -El costo por dosis -La frecuencia con la cual se debe administrar al cliente mascota
Resultados
Registrar la información de un medicamento recetado en el sistema.

Nombre	R. 5. 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. 6. 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. 7. 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. 8. 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	
<Ninguno>	
Resultados	
Dar de alta a un cliente mascota que haya estado hospitalizado	

Nombre	R. 9. Mostrar 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	
<Ninguno>	

Resultados	
Mostrar datos de hospitalización de un cliente mascota	

Nombre	R. 10. 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	
<Ninguno>	
Resultados	
Disponible el minicuarto que estaba siendo ocupado	

Nombre	R. 11. Mostrar ingresos por concepto de hospitalizaciones
Resumen	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	

Nombre	R. 12. Mostrar el número del mini cuarto que ocupa un cliente mascota hospitalizado
Resumen	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

Nombre	R. 13. Mostrar el historial de historias medicas de una mascota
Resumen	El programa debe mostrar el historial de historias medicas de una mascota
Entradas	
-ID del dueño -Nombre del dueño -Nombre de la mascota	
Resultados	
Mostrar el historial de historias medicas de una mascota	

REQUERIMIENTOS NO FUNCIONALES

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

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 clinicos, como crea historiales clinicos.

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

Página 12 de 16 - Algoritmos y Programación I - Universidad Icesi

		+getId():String +showMyinfo():String +showPetsinfo():String +retrieveNumberMiniRoom(String):int +getPet():String +getNumberOfMiniRoom():int +typeSelectionMssg():void +plainLine():void +locatePersonWithPet2AddStuff(String, String, String, int, String, String, ArrayList<ReqMed>):void +getName():String +getId():String +locatePet2AddStuff(String, int, String, String, ArrayList<ReqMed>):void +getName():String +addStuff2MedRec(int, String, String, ArrayList<ReqMed>):void +getSymptoms():String +setSymptoms(String):void +getDiagnosys():String +setDiagnosys (String):void +addNewMed(ReqMed):void
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	+menu():void +typeSelectionMssg():void +plainLine():void +showMenuOptions():void +showAllMedRecs4Pet():void

	ReqMed MedRec	+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 Person Pet MedRecord	+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. 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

<p>R. 13. Mostrar el historial de historias medicas de una mascota</p>	<p>Main Veterinary Person Pet MedRec ReqMed DateIn</p>	<pre>+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</pre>
--	--	--