UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Programación 1
Tarea Integradora 2 - Unidad 3

En la unidad 3 de nuestro curso hemos trabajado en el análisis de un problema construyendo un modelo con los elementos que intervienen en el problema y especificando los servicios que el programa debe ofrecer, bajo el paradigma de programación orientado a objetos.
Aprendimos a construir las clases que implementan el modelo de la solución del problema, identificando de manera informal los métodos de una clase y clasificarlos en métodos constructores, de consulta y de modificación. Utilizamos una arquitectura para un programa que permita repartir de manera adecuada las responsabilidades entre la interfaz de usuario y el modelo de la solución, y cómo relacionar dichos componentes. Finalmente, hemos aprendido a relacionar todos los conceptos vistos en las tres primeras unidades del curso.

Esta tarea integradora presenta una actividad en la cual se requiere aplicar todos los conocimientos adquiridos en la unidad 3. Por tanto, esta tarea es un instrumento para verificar el cumplimiento de los objetivos que han sido planteados para la unidad 3 descrita 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 (Lista de requerimientos funcionales en el formato visto en clase).
2. Diseño de la solución. Elabore un diagrama de clases que modele la solución del problema de acuerdo con las buenas prácticas y los patrones de diseño revisados hasta el momento en el curso. Su diagrama debe incluir el paquete modelo y el de interfaz de usuario.
3. Realice un diagrama de objetos que satisfaga su diagrama de clases.
4. 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.
5. Documentación en JavaDoc (Debe entregarse el JavaDoc generado y ubicarlo en la carpeta docs).
6. Usar GitHub como repositorio de código fuente y documentación utilizando la estructura de carpetas aprendida en clase. Recuerde que se debe evidenciar su avance a través de los días en el desarrollo de su tarea.
7. Subir a moodle los puntos anteriores el plazo máximo es 24 de Abril de 2021

Recuerde que puede encontrar la Rúbrica de la tarea integradora en el siguiente enlace.

UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Programación 1
Tarea Integradora 2 - Unidad 3

Nota:

- Usted debe entregar la URL de su repositorio GitHub 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:

  petCenter/
  src/
  bin/
  docs/
- Dentro de los directorios **src/** y **bin/** estarán presentes estos directorios(representando cada uno de sus paquetes):

  ui/
  model/
- El directorio src (source code) contiene sus clases .java dentro del directorio ui/ y model/. Por otro lado el directorio bin (binary files) contiene los archivos .class en el directorio ui/ y model/. El directorio docs tendrá toda la documentación de análisis y diseño
- Su código debería compilar de acuerdo con lo explicado en la diapositiva 15 de esta presentación: http://tinyurl.com/y3bd9bg2

A continuación, encontrará un enunciado que narra de forma detallada la situación problemática que se espera usted solucione.

Enunciado

Se quiere construir una aplicación que permita manejar las consultas de urgencias para un centro veterinario. El centro cuenta con un máximo de 7 veterinarios.

Diariamente las consultas de urgencias inician a las 6:30am y finalizan a las 10:30pm, llevando el registro de las mascotas atendidas, las cuales no pueden sobrepasar el máximo permitido (120 mascotas). Cada mascota tendrá asignado un estado de consulta así:

- Esperando ser atendido
- En consulta
- Traslado a hospitalización
- Salida autorizada
- Salida sin atención

Un veterinario sólo puede atender una mascota a la vez y solo si se encuentra en estado “Esperando ser atendido”.
Al cerrar el centro de urgencias,no deben existir mascotas esperando ser atendidas, generando las estadísticas del cierre diario, para luego proceder a eliminar la información de las mascotas atendidas.

UNIVERSIDAD ICESI

Departamento de TIC
Algoritmos y Programación 1
Tarea Integradora 2 - Unidad 3

El centro clasifica a los animales en 5 niveles de prioridad de acuerdo al tiempo de espera para ser atendido:

- ***Prioridad 1*** o ***rojo***: necesita atención de forma inmediata.
- ***Prioridad 2*** o ***naranja***: el tiempo de espera es máximo 10 minutos.
- ***Prioridad 3*** o ***amarillo***: la atención puede demorarse 60 minutos.
- ***Prioridad 4*** o ***verde***: la atención puede demorarse 2 horas.
- ***Prioridad 5*** o ***azul***: la atención puede demorarse 4 horas.

Al ingresar al centro, una mascota se registra en un nivel de prioridad con la siguiente información: especie (perro, gato, conejo, reptil, pájaro), nombre, edad, raza (para perro y gato), su dueño y sus síntomas. Automáticamente la mascota queda en espera de ser atendido.
Del dueño se tiene su número de identificación, nombre, celular y dirección.
Nota: un nombre de mascota y un nombre del dueño sólo puede existir una vez. Es decir que, en el centro de urgencias solo podrá existir una sola vez la combinación (nombre mascota y nombre dueño).

El sistema debe indicarle al veterinario cual es la próxima mascota que atenderá e iniciar la consulta. Esta mascota es seleccionada según su orden de llegada y el nivel de prioridad al que pertenece, seleccionando a la primera mascota encontrada sin atender buscando desde la prioridad más alta.

Una vez que la consulta ha terminado, el veterinario debe indicarle al sistema el fin de esta, quedando disponible para atender otros animales.

Una mascota puede abandonar el centro veterinario en cualquier momento, siempre y cuando no se encuentre en consulta.

La aplicación deberá permitir realizar las siguientes actividades:

1) Adicionar un veterinario con la siguiente información: número de identificación, nombre, apellidos y registro único veterinario. En caso que se haya llegado al máximo permitido debe notificar el error.

2) Eliminar un veterinario siempre y cuando no existan mascotas registradas en el centro

3) Registro de una mascota al centro

4) Retiro de una mascota del centro siempre y cuando, la mascota se encuentre pendiente de ser atendida.

5) Iniciar una consulta indicando la cédula del veterinario que atenderá al paciente, para lo cual el veterinario debe existir y no podrá tener una mascota en consulta; en caso contrario deberá informar del error. El sistema buscará la siguiente mascota a atender y en caso de no encontrar ninguna mascota, deberá informar al usuario. En este momento la mascota cambiará su estado por “en consulta” y registrará al veterinario que la atenderá.

6) Finalizar una consulta indicando la cédula del veterinario, el nombre de la mascota y si autoriza la salida o si por el contrario debe pasar a hospitalización. En caso que la cédula no corresponda a un veterinario o que la mascota no se encuentre en consulta con el veterinario debe notificar del error. Recuerde que

el veterinario quedará disponible para atender una mascota y que la mascota conservará el veterinario que la atendió.

7) Mostrar el número de mascotas que no han sido atendidas.

8) Cierre diario del centro:
    a. Verificar que no existan mascotas pendientes por atender
    b. Generar la información estadística:
        i. Mostrar el nombre del veterinario con mayor número de consultas
        ii. Cantidad de mascotas atendidas por prioridad
        iii. Porcentaje de mascotas que salieron del centro sin ser atendidas
    c. Eliminar todas las mascotas atendidas