**Trabajo Práctico Grupal Final**

**Programación y Laboratorio 3**

Universidad Tecnológica Nacional – Facultad Regional Mar del Plata

**Integrantes:**

Duarte Nicolás - Mirabella Nahuel - Silva Nieva Jairo

Introducción:

Con las consignas planteadas para el trabajo práctico integrador decidimos que sería una buena idea poder aplicar lo aprendido para realizar un sistema que permita la gestión de una Veterinaria.

Decidimos que principalmente el sistema contaría con registro de empleados y de clientes. A su vez quedarían registradas la/las mascotas de cada cliente. También se podrían asignar turnos a los clientes y luego facturar por los servicios brindados.

Gracias a que uno de nosotros trabaja en una veterinaria, pudimos plantear una lista de requisitos básicos que el sistema debería cumplir.

Estos inicialmente fueron:

- Agregar a un registro nuestros clientes

- Agregar mascotas para el cliente

- Modificar datos del registro

- Validaciones para la creación y modificación de clientes

- Eliminar datos del registro

- Programar turnos

- Listar próximos turnos

- Atender (depende de insumos disponibles)

- Realizar historia clínica luego de ser atendido

- Ver historia clínica de una mascota

- Facturar por servicio brindado

Métodos con validación de stock y de personal:

-método consulta

-método urgencia

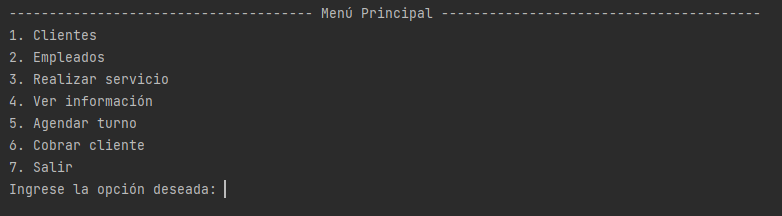
-método internación

-método cirugía

-método castración

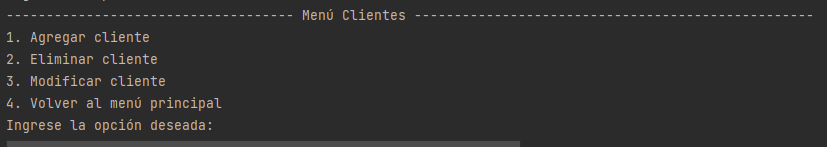
Informe técnico:

Este proyecto busca facilitar la gestión de registros de una veterinaria, decidimos diseñar un sistema que cuenta con un menú para su utilización. Desde ahí se puede acceder a los diferentes sub-menú en los que decidimos dividir las tareas del sistema.



Las dos clases que heredan de la clase abstracta Persona y que consideramos base fundamental de nuestro programa son Clientes y Empleados. De estas dos están compuestos los principales registros de la clase contenedora Veterinaria.

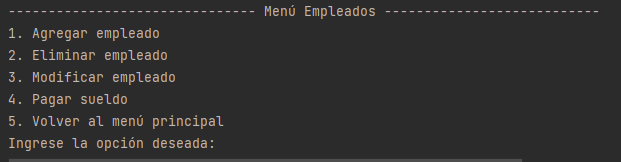
En los dos primeros sub-menú podremos ver todo lo que respecta a Alta, Baja y Modificación de los mismos.



Los Clientes contienen listas de sus mascotas, ya sean Gatos y/o Perros. Y estos a su vez contienen un registro que denominamos Historia Clínica, en donde se deja constancia de los servicios realizados para cada mascota.

También cuentan con registro de Tickets en donde se muestra la información de servicios anteriores realizados.

El sub-menú de Empleados además de tener las opciones de ABM cuenta con una cuarta opción en la que nos permite pagarle el sueldo a un Empleado (este depende los servicios realizados).



La clase Empleado es una clase padre de la cual heredan sus dos hijas: Veterinario quien realiza los servicios que requieren las mascotas y Cajero quien se encarga de realizar los cobros a los Clientes.

Tanto Cliente, como Empleado tienen varias comprobaciones a la hora de darlos de alta en el registro. Desde que no dejen ningún atributo sin completar, hasta que se respeten formatos tanto en fechas como en e-mail.

Después los siguientes sub-menú están orientados a:

La visualización de los registros: ya sea de clientes o de empleados.

La asignación de turnos: en donde queda registrado el cliente, el día, la hora y el veterinario que lo va a atender.

El cobro a un cliente: se realiza un ticket por los servicios prestados.

Por último, para lograr persistencia en nuestros registros son guardados en archivos. Toda la información es cargada antes de iniciar el programa y se guarda al salir del mismo.

Durante el desarrollo del sistema lo que más dificultades nos trajo fue la toma de decisiones a la hora del estructurado de clases, que atributos debería tener cada clase, como se relacionarían entre sí empleados y clientes. También fue trabajo de varios días pensar cómo debería estar organizado el menú y luego, una vez planteado, lograr dejarlo completamente funcional.

Fuentes de Información consultadas:

* Todo el material disponible en las presentaciones de la cátedra de Programación y Laboratorio 3 de la Universidad Tecnológica Nacional brindada por el docente Leonardo Chiessa.
* <https://docs.oracle.com/javase/tutorial/essential/exceptions/handling.html>
* <https://docs.oracle.com/javase/tutorial/essential/exceptions/throwing.html>
* <https://docs.oracle.com/javase/7/docs/api/java/io/File.html>
* <https://docs.oracle.com/javase/7/docs/api/java/io/Serializable.html>
* <https://docs.oracle.com/javase/8/docs/api/java/beans/Transient.html>
* <https://www.oracle.com/technetwork/developer-tools/jdev/gettingstartedwithumlclassmodeling-130316.pdf>