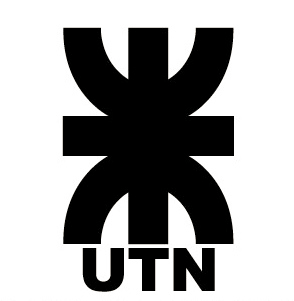
**Algoritmos y Estructuras de Datos**



**Universidad Tecnológica Nacional**

**Facultad Regional Tucumán**

***TECHNO DIGITAL***

**Comisión: 1k11**

**Integrantes:**

* Giménez Leandro Nahuel **Legajo:**122803

[Leandro.Gimenez@alu.frt.utn.edu.ar](mailto:Leandro.Gimenez@alu.frt.utn.edu.ar)

* Garcia Jose Elias **Legajo:** 52279

[Jose.Garcia@alu.frt.utn.edu.ar](mailto:Jose.Garcia@alu.frt.utn.edu.ar)

**Consignas**

**Implementación de Módulos**

**Módulo Consultorio:**

Esta parte del proyecto estará dirigida a los diferentes profesionales que atienden y registran la evolución de las mascotas en una base de datos de historias clínicas. El sistema deberá contar con un listado de las mascotas que están registradas para ser atendidas ese día, para visualizarla, el veterinario deberá identificarse en el sistema por medio de su número de matrícula y contraseña y accediendo a la opción indicada. De esta manera el veterinario llamará por apellido y nombre a la mascota que atenderá, una vez realizada esta tarea procede al ingreso de la evolución, en un texto de no más de 380 caracteres. Luego de realizada esta operación la mascota debe desaparecer del listado.

Los datos de interés para el veterinario en este proceso son: Apellido y Nombres de la mascota (el apellido corresponde al dueño o familia), DNI del dueño, Localidad, Edad (calculada con la fecha de nacimiento registrada), Peso.

La historia clínica de la mascota es común para todos los veterinarios, por lo tanto, se deberá visualizar fecha de atentación de la misma y el nombre del veterinario que la redacto.

**Módulo Recepción:**

Este módulo satisface las necesidades del personal que asiste a los veterinarios en la atención al público. Desde aquí se hace ingreso de las mascotas, y la registración de los turnos.

Esta aplicación debe permitir obtener un informe de las mascotas atendidos en determinada fecha por un determinado veterinario.

**Módulo Administración:**

La gerencia del centro veterinario es la encargada de realizar el alta de los veterinarios que trabajan en la institución, así como también de los empleados que realizan la registración de los turnos y mascotas.

Es el área encargada desea visualizar las atenciones realizadas por los profesionales según las en el mes.

Para incentivar a los veterinarios, la gerencia otorga un bono mensual al profesional que haya registrado la mayor cantidad de turnos en ese periodo.

**I. INTERFAZ DE USUARIO**

El diseño visual de la interfaz es libre y se pueden utilizar elementos gráficos (logos, colores) para hacer las pantallas de la aplicación más atractivas. Por ejemplo, utilizar símbolos o íconos para representar las especialidades de los médicos, las pantallas de los informes, etc. Opcionalmente, se le podrá dar un nombre y proponer un logo que represente a la aplicación y utilizarlo en las pantallas.

**Modulo Consultorio Veterinario**

=========================

1.- Iniciar Sesión

2.- Visualizar Lista de Espera de Turnos (informe)

3.- Registrar Evolución de la Mascota

4.- Cerrar la aplicación.

Ingrese una opción: \_

**Modulo del Asistente**

=========================

1.- Iniciar Sesión

2.- Registrar Mascota

3.- Registrar Turno

4.- Listado de Atenciones por Veterinario y Fecha

5.- Cerrar la aplicación.

Ingrese una opción: \_

**Módulo Administración**

=========================

1.- Registrar Veterinario

2.- Registrar Usuario Asistente

3.- Atenciones por Veterinarios

5.- Ranking de Veterinarios por Atenciones

6.- Cerrar la aplicación.

Ingrese una opción: \_

**II. GESTIÓN Y AUTENTICACIÓN DE USUARIOS**

Dentro del módulo administración, el usuario tendrá la posibilidad de crear un nuevo veterinario y la opción de crear un nuevo asistente. Para esto el programa le solicitará nombre de usuario y contraseña.

**Nombre de usuario:** Quedará definido por una cantidad mínima de 6 caracteres y máxima de 10, los cuales podrán ser letras, números y/o símbolos del conjunto {+, -, /, \*?,,¿,!,¡}. Deberá cumplir con los siguientes requisitos.

a. Ser único para cada usuario registrado.

b. Comenzar con una letra minúscula.

c. Tener al menos 2 letras mayúsculas.

d. Tener como máximo 3 dígitos.

Ejemplos de nombres de usuario incorrectos: AbC123 (no cumple con b), pTS!1234 (no cumple con d), g178Mci (no cumple con c), mARtin123gomez (tiene más de 10 caracteres).

Ejemplos de nombres de usuario correctos: mARtin12, jo97! AR.

**Contraseña:** Su conformación no podrá darse al azar, sino que deberá respetar lo siguiente:

a. Deberá contener al menos una letra mayúscula, una letra minúscula y un número.

b. No podrá contener ningún carácter de puntuación, ni acentos, ni espacios. Sólo caracteres alfanuméricos.

c. Deberá tener entre 6 y 32 caracteres.

d. No debe tener más de 3 caracteres numéricos consecutivos.

e. No debe tener 2 caracteres consecutivos que refieran a letras alfabéticamente consecutivas (ascendentemente). Este criterio es válido tanto para letras mayúsculas, minúsculas o combinación de ambas.

Ejemplos de contraseñas mal formadas: Ach32 (no cumple con c), dorit1234 (no cumple la regla a ni la d), sA;gotAP.10 (no cumple con b), aBuel123 (no cumple con e).

Ejemplo de contraseñas bien formadas: Achus32, 125Af89, aLejo123, DORItO45, 4aC2sa.

Nota: Tanto en el nombre de usuario como en la contraseña deben distinguirse mayúsculas y minúsculas.

**III. PERSISTENCIA**

Para no perder la información entre ejecuciones, es necesario mantener los datos de las cuentas de usuario registradas. Para cumplir con este objetivo deberá implementarse la persistencia de la información manejando archivos que se describirán más adelante en este apartado. La estrategia sugerida para cumplir con este objetivo es:

● Cuando se inicia la aplicación, se cargan los datos en estructuras de datos en memoria.

● Durante la ejecución de la aplicación se actualizan estas estructuras.

● Al terminar cada sesión se agrega el contenido de la misma al archivo correspondiente.

Los datos de las cuentas de usuario se almacenarán en los archivos Usuarios.dat y Veterinarios.dat. Estos archivos serán de formato binario y guardarán variables diseñadas en una estructura de datos a fin de representar la información asociada a las cuentas de usuarios (es decir, nombre y contraseña). Cada vez que se inicie la aplicación, se utilizará la información almacenada en este archivo para validar el inicio de sesión y/o la creación de un nuevo usuario.

Cada vez que un usuario cree un registro de mascota el mismo se guardará dentro del archivo Mascotas.dat y los turnos correspondientes dentro del archivo Turnos.dat. Se sugiere utilizar el siguiente diseño tanto para los archivos mencionados, como para las estructuras en memoria que donde se registran los procesos descritos:



**Introducción**

El siguiente trabajo consta del analisis y seguimiento del desarrollo de consignas propuestas por la catedra de algoritmo y estructuras de datos que se detallaran a continuación, este trabajo esta propuesto para integrar los diversos temas y conocimientos adquiridos durante el cursado del año 2020.

El trabajo fue gestionado mediante la herramienta GitHub/GitHub desktop el cual facilita el trabajo en grupo por medio de repositorios para trabajar de manera más eficaz y fácil. El repositorio GitHub es el siguiente:

<https://github.com/Elias99garcia/Trabajo-integrador-segundo-cuatrimestre>.

La consigna de este trabajo integrador de segundo cuatrimestre se basa en desarrollar una aplicación que permita al usuario registrar atención de una veterinaria para mascotas, con múltiples consultorios de distintos profesionales. Además, deberá constar tambien de un módulo para la gestión de turno y otro para la administración gerencial. Cuyos módulos son los siguientes.

El trabajo incluye todos los conceptos aprendidos a lo largo de la materia Algoritmo y estructura de datos, cursada actualmente en 2020, desde los conceptos básicos aprendidos en primeras instancias de C++ hasta los últimos temas aprendidos, funciones, archivos multidimensionales, cadenas, registros y archivos binarios,

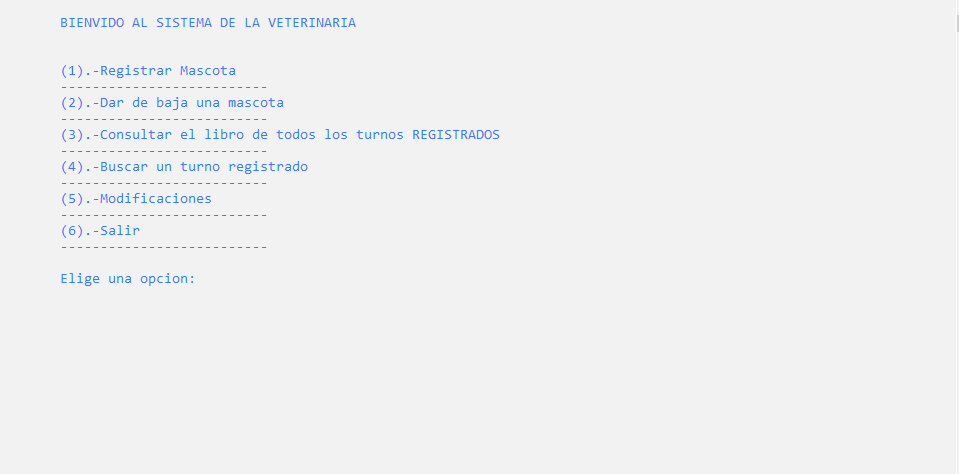
**Modulo consultorio:** Esta parte se encarga de gestionar las evoluciones de las mascotas además de corroborar los turnos de los pacientes, claro, primero para esto se debe iniciar sección del veterinario que esté operando.

El modelo de menú usado fue el siguiente:



**Módulo recepción/asistente:** Este módulo está encargado de atender y satisfacer las necesidades del veterinario para la atención al cliente. Este se encarga de registrar las mascotas, los turnos, y además ver un listado de los turnos con su respectiva fecha y veterinario que lo atienda.

El modelo de menú usado es el siguiente:



**Modulo administración:** Este módulo se encarga de la parte administrativa del programa el cual consta de: registrar un veterinario, registrar un asistente de veterinario, ver una lista de atenciones realizadas por el veterinario y además ver un ranking de veterinarios organizados por cantidad de atenciones.

El modelo de menú utilizado fue el siguiente:

