



Segundo Trabajo Integrador

Cátedra de Algoritmos y Estructura de Datos.

Integrantes:

* Paz Posse Tobías
* Pérez Lucas Gustavo
* Paz Bruno Leonel
* Posse Ricardo

Comisión: 1K5

# 

# **Introducción**

El siguiente trabajo consta del análisis y seguimiento del desarrollo de las consignas propuestas por la cátedra de Algoritmos y Estructura de Datos, en un intento de integrar los diversos conocimientos adquiridos durante su cursado, por medio de la implementación de un conjunto de programas interconectados con menús y archivos de datos de acceso compartido. El código aquí fue escrito tratando de cumplir con las buenas prácticas de la programación que se vieron durante el año. Además, su armado fue gestionado, controlado y compartido, por medio de la herramienta de control de versiones Git y la plataforma GitHub, siendo el uso de estas últimas facilitado por la colaboración de los mentores de la comunidad ProgramaConNosotros.

La consigna de este segundo trabajo integrador se basa en desarrollar una aplicación que permita registrar las atenciones de las mascotas que asisten de un centro veterinario con múltiples consultorios de distintos profesionales. Deberá constar también de un módulo para la gestión de turno y otro para la administración gerencial. Cuyos módulos son los siguientes:

1. **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), y el Peso- La historia clínica de la mascota es común para todos los veterinarios, por lo tanto, se deberá visualizar fecha de atención de la misma y el nombre del veterinario que la redactó.

1. **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.

1. **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 que 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.

# **Menú principal**

***Diseño:***

Veterinaria C Mais Mais

.......................

1. Ingresar como Administrador

2. Ingresar como Veterinario

3. Ingresar como Asistente

4. Salir del programa

>

Elegimos este diseño en base al modelo propuesto por la Catedra de Algoritmo y Estructura de datos, pero con ediciones propios del grupo. Decidimos crear un menú principal donde cada usuario tenga la oportunidad previa de decidir cómo quiere ingresa y luego sea designado a su módulo correspondiente.

***Funcionamiento:***

El menú funciona a través de un ciclo ***Do While*** cual contiene una estructura ***switch***. Dentro de los 3 casos posibles se pide que el usuario ingrese matricula o nombre de usuario, dependiendo como desee ingresar. A partir de ello se verifica dentro de una estructura ***If*** si los datos ingresados coinciden con los datos en la “***LibreriaModulos.h",*** Luego se pide que ingresen la contraseña cual también es verificada con todos sus parámetros. Finalmente, el usuario es designado a su módulo correspondiente dela siguiente manera:

**system("ModuloAdministracion.exe");**

**system("ModuloConsultorioVeterinario.exe");**

**system("ModuloAsistente.exe")**

# **Módulo Consultorio**

***Diseño:***

Con respecto al diseño, optamos por seguir al modelo que propusieron la Cátedra de Algoritmos y Estructura de Datos.

**Modulo Consultorio Veterinario**

**=========================**

**1.- Visualizar Lista de Espera de Turnos (informe)**

**2.- Registrar Evolución de la Mascota.**

**3.- Cerrar sesión.**

**>**

Un diseño que es simple de hacer y bastante intuitivo. Podemos observar que, en este módulo, el doctor puede acceder a los turnos, tanto los disponibles como los ya atendidos. Luego, en una segunda opción, vemos que el veterinario puede registrar la evolución de una mascota determinada, si esta sigue bajo observación o si ya se le otorgó la alta médica. Finalmente, decidimos en grupo quitar la opción de inicio de sesión ya que lo haríamos antes de usar el modulo. El inicio de sesión se hace en un menú donde se elige quien desea iniciar y a partir de ello se lo lleva al módulo que perteneces ese usuario.

***Funcionamiento:***

1. Visualizar Lista de Espera de Turnos:

Utilizamos una función sin tipo cuyos parámetros son los archivos “*Veterinarios.dat”* y *“Turnos.dat”*. Dentro de esta función tendremos 2 banderas:

**matEncontrada** = Nos indica si la matricula fue encontrada en el sistema.

**matValida** = Nos indica si la matricula del veterinario se encuentra registrada en el sistema o no.

Con la ayuda de un ciclo ***Do While*** le pedimos al usuario que ingrese la matricula, acto seguido con una función *booleana* que está en la librería ***“LibreríaModulos.h”*** verificamos si, la matricula ingresada, se encuentra ya registrada o no. Luego con un ***if*** abrimos los dos archivos de la siguiente manera:

**archi = fopen (“Veterinarios.dat”, “r+b”);**

**archi2 = fopen (“Turnos.dat”, “r+b”);**

Después de abrir ambos archivos, toca leerlos, para ello necesitamos realizar la próxima función:

**fread(&read, sizeof(turnos), 1, archi);**

**fread(&read, sizeof(veterinario), 1, archi2);**

Una vez leído el archivo, con un ciclo ***While*** recorremos todo, desde inicio a fin, el archivo *“Turnos.dat”* e imprimimos en pantalla, con la ayuda de un ***if***, los turnos pendientes y realizados del veterinario. Y con otro ciclo ***While*** y un ***if*** verificamos si la matricula ingresada se encuentra dentro de los registros.

Al final de la función tendremos que cerrar ambos archivos, para ello haremos lo siguiente:

**fclose(archi);**

**fclose(archi2);**

1. Registrar Evolución de la Mascota.

En esta opción utilizaremos una función sin tipo cuyos parámetros son los archivos *“Mascotas.dat”* y *“Turnos.dat”.* Realizamos los mismos pasos que en la opción anterior, leemos ambos archivos y con un ciclo ***Do While*** le pedimos al usuario que ingrese el Apellido y Nombre de la mascota, acto seguido con la función ***strcmp*** comparamos el nombre ingresado con los nombres guardados dentro del archivo *“Mascotas.dat”*:

**(strcmp(regi.nombre, apeynom) == 0)**

Si esto se cumple, el usuario ingresa el informe y lo guardamos en un ***aux***. Acto seguido copiamos el contenido del ***aux*** al apartado ***atención*** del registro ***turnos*** y con la siguiente función guardamos el informe

**fwrite(&reg, sizeof(reg),1,archi2);**

# **Módulo Asistente**

***Diseño:***

En este módulo también optamos por seguir al modelo que propusieron la Cátedra de Algoritmos y Estructura de Datos.

**Módulo de Asistente**

**===================**

**1. Registrar mascota**

**2. Listar mascotas**

**3. Registrar turno**

**4. Listar turnos por veterinario y fecha**

**5. Cerrar Sesión**

**>**

Este diseño es igual al anterior donde quitamos el inicio de sesión para hacerlo anterior al ingreso del módulo. En este módulo se puede observar que el asistente tiene la posibilidad de registrar a una mascota (1°opcino) o registrar un turno para las mascotas (3°opcion). El usuario asistente también puede ver la lista de mascotas (2°opcion) con todos sus datos previamente registrados. Finalmente, tiene su última función donde puede ver la lista de turnos según el veterinario y la fech.

***Funcionamiento:***

1. Registrar mascota:

Para trabajar en esta función abrimos el archivo “mascotas.dat”, donde registraríamos todos los datos de la mascota a registrar. Para ello se pide nombre, dirección, DNI, localidad, peso, teléfono y edad.

**archi = fopen("Mascotas.dat", "r+b");**

Para poder calcular la edad como es pedido se hace uso de la librería ***“time.h”***y se pide ingresar la fecha de nacimiento para poder llevar acabo la siguiente función:

**time\_t t** = time(NULL);

**struct tm tm** = \*localtime(&t);

**anios** = (float)(((tm.tm\_year + 1900) \* 365) + ((tm.tm\_mon + 1) \* 30) + (tm.tm\_mday) - ((reg.fec.anio) \* 365) - ((reg.fec.mes) \* 30) - (reg.fec.dia)) / 365;

**reg.edad** = anios;

Una vez finalizada esta función y cargados todos los datos debemos escribirlos en el archivo y finalmente cerrarlos para concluir con la función. Para esto se hace uso de:

**fwrite(&reg, sizeof(reg), 1, archi);**

**fclose(archi);**

1. Listar Mascota.

El objetivo de este módulo es mostrar en pantalla todas las mascotas registradas con sus datos. Lo primero que hacemos es abrir el archivo donde habíamos registrado la mascota anteriormente.

**archi = fopen ("Mascotas.dat", "r+b");**

Una vez abierto el archive llamaremos a la función debemos leer el archivo, para eso usamos la función:

**fread(&reg, sizeof(reg), 1, archi);**

Luego usaremos una estructura ***while (!feof(archi))*** para recorrer el archivo continuamente devolviendo la información dentro del archivo hasta que no se encuentren más mascotas registradas.

Por cada mascota que se muestre en pantalla tendremos un contador ***“contadorAnimales++;”*** de esta forma, si no el contador es igual a cero el programa nos devolverá un mensaje diciendo que no hay mascotas registradas.

Para finalizar la función se debe cerrar el archivo.

**fclose(archi);**

1. Registrar turno

Para esta función debemos abrir tres archivos de la siguiente manera:

**archi = fopen("Turnos.dat", "r+b");**

**archi2 = fopen("Veterinarios.dat", "r+b");**

**archi3 = fopen("Mascotas.dat", "r+b");**

Una vez que los archivos están abiertos, solicitamos ayuda de un ciclo ***Do While*** donde le pedimos al usuario que ingrese la matricula del veterinario, acto seguido con una función *booleana* que está en la librería ***“LibreríaModulos.h”*** verificamos si, la matricula ingresada, pertenece a un veterinario ya registrada o no. Luego pedimos al usuario que registre su turno.

Una vez que el usuario registro la fecha de su turno se hace uso de un ciclo ***Do While*** donde le pedimos al usuario el ingreso de su DNI, luego leemos el archivo con la siguiente función:

**fread(&masc, sizeof(mascota), 1, archi3);**

Paso siguiente verificamos si el DNI ingresado coincide con el DNI de un usuario registrado, de ser así se termina de ingresar los datos y finalmente se cierran los archivos.

**fclose(archi);**

**fclose(archi2);**

**fclose(archi3);**

1. Listar turnos por veterinario y fecha

Utilizamos una función sin tipo cuyos parámetros son los archivos “*Veterinarios.dat” y “Turnos.dat”.* Dentro de esta función tendremos 2 banderas:

**matEncontrada** = Nos indica si la matricula fue encontrada en el sistema.

**matValida** = Nos indica si la matricula del veterinario se encuentra registrada en el sistema o no.

Con la ayuda de un ciclo ***Do While*** le pedimos al usuario que ingrese la matricula, acto seguido con una función *booleana* que está en la librería ***“LibreríaModulos.h”*** verificamos si, la matricula ingresada, se encuentra ya registrada o no. Luego abrimos los tres archivos de la siguiente manera:

**archi3 = fopen("Mascotas.dat", "r+b");**

**archi2 = fopen("Veterinarios.dat", "r+b");**

**archi = fopen("Turnos.dat", "r+b");**

Una vez encontrada la matricula se abre la lista de turnos del veterinario a quien pertenece esa matricula. Luego se hace uso de un ciclo ***Do While*** donde el usuario registra la fecha que desea buscar (mes y año). Esta fecha finalmente es determinada con ayuda de una estructura ***switch*** de la siguiente manera:

 switch (mes)

    {

    case 1:

        printf("enero");

        break;

    case 2:

        printf("febrero");

        break;

    case 3:

        printf("marzo");

        break;

    case 4:

        printf("abril");

        break;

    case 5:

        printf("mayo");

        break;

    case 6:

        printf("junio");

        break;

    case 7:

        printf("julio");

        break;

    case 8:

        printf("agosto");

        break;

    case 9:

        printf("septiembre");

        break;

    case 10:

        printf("octubre");

        break;

    case 11:

        printf("noviembre");

        break;

    case 12:

        printf("diciembre");

        break;

    }

Paso siguiente, se lee el archivo ***“Turnos.dat”*** con la funciona ***fread*** y se lo recorre continuamente con la función ***while (!feof(archi***) para verificar que la fecha coincida con un turno. Finalmente, se muestra el turno en pantalla con los detalles de la evolución de la mascota (si es que no fue dado de alta o esta en espera) y cerramos los archivos

**fclose(archi);**

**fclose(archi2);**

**fclose(archi3);**

# **Módulo Administración**

***Diseño:***

En este módulo a diferencia de los anteriores tomamos un modelo diferente propuesto por nosotros.

Módulo Administración

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

1. Registrar Veterinario

2. Registrar Usuario Asistente

3. Registrar Administrador

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

4. Ranking de Veterinarios por Atenciones

5. Atenciones por Veterinarios

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

6. Cerrar sesion

>

Este diseño permite registrar usuarios (veterinario, asistentes o administrador) quienes tendrán acciones limitadas, cada uno tendrá acceso su módulo especifico, prohibiéndoles ingresar a otros módulos.

El administrador es el encargado de este módulo quien tendrá la posibilidad de crear más usuarios. Otra función a la que tendrá acceso el administrador es a ver el ranking de atención de veterinarios. Por último, este módulo le permite ver los turnos que ha atendido cada veterinario de este modo podrá supervisar quien cumple con su trabajo.

***Funcionamiento:***

1. Registrar Veterinario:

Pedimos al administrador que ingrese una matricula para el nuevo veterinario. Usamos una función booleana para determinar si la matricula es valida. En caso de no ser asi el programa notificara la situación y si cumple los parámetros.

Luego se pide al usuario que ingrese su contraseña, para verificar que esta este ingresada correctamente se utiliza una función de la librea ***“LibreiaModulos.h”,*** de cumplir con todos los requisitos se ingres los datos correspondiente y se escribe y cierra el archivo de la siguiente manera:

**fseek(arch1, 0, SEEK\_END);**

**fwrite(&reg, sizeof(reg), 1, arch1);**

**fclose(arch1);**

1. Registrar Usuario Asistente:

Pedimos al administrador que ingrese un nombre de usuario para el nuevo Asistente. Usamos una función booleana para determinar si el nombre es valido. En caso de no ser asi el programa notificara la situación y si cumple los parametros.

Luego se pide al usuario que ingrese su contraseña, para verfiicar que esta este ingresada correctamente se utiliza una función de la librea ***“LibreiaModulos.h”,*** de cumplir con todos los requisitos se ingres los datos correspondiente y se escribe y cierra el archivo de la siguiente manera:

**fseek(arch1, 0, SEEK\_END);**

**fwrite(&reg, sizeof(user), 1, arch1);**

**fclose(arch1);**

1. Registrar Administrador:

El administrador se genera de la misma manera que el usuario y tiene los mismos privilegios que el administrador creando la cuenta. A la hora de llamar la función en el main aparece la diferencia .

**RegistrarUsuario(arch2, 1, ArchivoUsuarios); //El argumento 1 indica que es un administrador**

**RegistrarUsuario(arch2, 3, ArchivoUsuarios); //El argumento 3 indica que es un asistente**

1. Ranking de Veterinarios por Atenciones:

Para operar en esta función se deberá abrir los archivos de la siguiente manera:

**arch1 = fopen (archiVeterinarios, "r+b");**

**arch2 = fopen (archiTurnos, "r+b");**

Una vez abierto los archivos haremos uso de una estructura ***while*** donde el archivo se leerá continuamente buscando los turnos por veterinario. Para poder lograr esto se comparan dentro de una estructura ***if*** las matriculas guardadas de los veterinarios y turnos, de ser iguales y cumplir con su turno se usará un contador de la siguiente manera:

**while (!feof(arch2))**

**{**

**if (infoVet.matricula == infoTurno.matriculaVet and infoTurno.borrado == true)**

**{**

**atendidos++;**

**}**

**fread(&infoTurno, sizeof(turnos), 1, arch2);**

**}**

Luego, se guarda el nombre del veterinario, su matrícula, y el contador de turnos atendidos en una variable del tipo registro rank (con esos 3 campos) y se llama a una función que inserta este registro en forma de nodo en una lista dinámica. La lista dinámica tiene una inserción de nodos de forma ordenada en función de la cantidad de turnos atendidos, es decir que lo único que hace falta es cargar la lista, el ranking toma forma automáticamente.

Finalmente se recorre la lista y se muestra en pantalla el ranking de veterinarios que mas turnos atendieron.

1. Atenciones por Veterinario

Con la ayuda de un ciclo ***Do While*** le pedimos al usuario que ingrese la matricula, acto seguido con una función *booleana* que está en la librería ***“LibreríaModulos.h”*** verificamos si, la matricula ingresada, se encuentra ya registrada o no. De ser asi se abre el archivo ***“turnos.dat***” de la siguiente manera:

**arch2 = fopen(archiTurnos, "r+b");**

Una vez abierto el archive se emplea una lectura y el uso del ciclo ***while*** cual tendrá una estructura ***if*** que verifique cuantos turnos atendió correctamente el veterinario seleccionado anteriormente por la matricula, este será ejecutado de la siguiente manera:

**while (!feof(arch2))**

**{**

**if (infoTurno.matriculaVet == matricula and infoTurno.borrado == true)**

**{**

**contadorTurnosAtendidos++;**

**printf("\n%d\_", contadorTurnosAtendidos);**

**printf("\t%20s: %d/%d/%d\n", "Fecha", infoTurno.fec.dia, infoTurno.fec.mes, infoTurno.fec.anio);**

**printf("\t%20s: %d\n", "Dni duenio", infoTurno.DNIduenio);**

**printf("\t%20s: %s\n", "Observaciones", infoTurno.atencion);**

**}**

**fread(&infoTurno, sizeof(turnos), 1, arch2);**

}

Finalmente, se mostrará en pantalla el número de turnos atendidos por el veterinario que el usuario haya decidido buscar según su número de matrícula.