



Universidad Autónoma de
Nuevo León

Programación Avanzada

Manual de usuario

NOMBRE: Eliud Asael Sánchez Ávila

Matrícula: 2062721

NOMBRE: Kevin Felipe Palacios Muñiz

Matrícula: 1873855

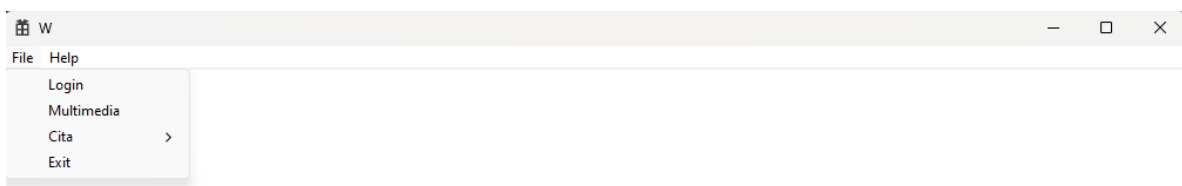
Profesor: Luisa Alejandra Monreal Méndez

Grupo: 064

INDICE	Página
Inicio	3
Login	3
Agregar Cita	4
Lista de Citas	5
Información del doctor	6
About	6

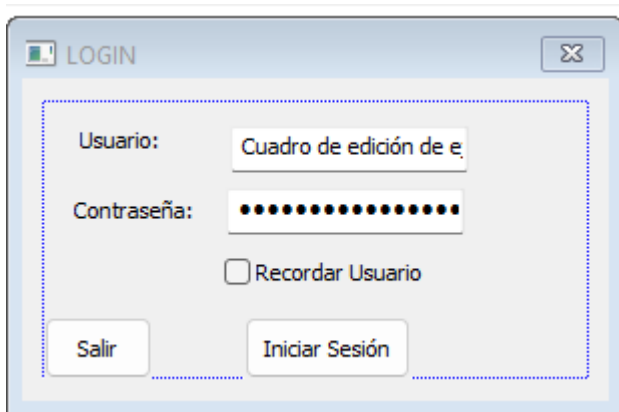
INICIO

Al inicio, se le desplegará una ventana del menú, donde vendrán las opciones de funciones para la página de citas. Dentro de File, vendrán las opciones principales del programa las cuales son: Login para iniciar sesión, multimedia, cita; la cual es para la generación y manejo de citas, y exit; para salir y que le guarde todo en un archivo binario.



Cabe mencionar que la sección de Help trae consigo las opciones de información del doctor y otra llamada about acerca de la información del programa.

LOGIN



Esta ventana es la esencial para que el programa funcione, como solo es un usuario el que usa el programa, automáticamente debe darle click al botón Iniciar Sesión para desbloquear las demás funciones del sistema.

Agregar Cita

Cuando le de a la sección de “agregar” en el apartado de cita, le desplegará la pantalla siguiente:

Dr. Santiago Jimenez

Nombre del cliente: Cuadro de edición de ejemplo

Nombre de la mascota: Cuadro de edición de ejer

Telefono: Cuadro de edición de ejemplo

Motivo de la consulta: Cuadro de edición de eje

Especie: ☐ Perro ☐ Gato ☐ Conejo

Costo: Cuadro de edición de ejemplo

Fecha: 23/11/2023

Estatus: Cuadro de edición de ejemplo

Estático Leer Fecha

Examinar

Static

Regresar

Agregar Cita

Aquí, podrá ingresar los datos en los cuadros de edición, cabe mencionar que, al acceder aquí, deberá borrar los mensajes incluidos en los cuadros. Cuando haya llenado todos los campos, le da a agregar cita, la cual guardará los datos de la cita en una lista que posteriormente será mostrada.

LISTA DE CITAS

Esta es la pantalla “principal” del programa donde se encuentra la mayor parte del manejo de las listas usadas. Para acceder a esta pantalla debe seleccionar el apartado de “Lista” ubicado en la sección “Cita” del menú principal.

MOSTRAR CITAS

Dr. Santiago Jimenez

Actualizar Cita Leer Cita

LISTA DE CITAS

Especie: ☐ Perro ☐ Gato ☐ Conejo

Telefono: Cuadro de edición de ejemplo

Nombre de la mascota: Cuadro de edición de ejemplo

Motivo: Cuadro de edición de ejemplo

Costo: Cuadro de edición de ejemplo

Estatus: Cuadro de edición de ejemplo

23/11/2023

Eliminar Cita

Static

Leer Fecha

Foto:

noviembre de 2023

dom.	lun.	mar.	mié.	jue.	vie.	sáb.
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	1	2
3	4	5	6	7	8	9

Aquí, podrá visualizar las citas creadas en el cuadro de la izquierda, las cuales vienen con el nombre del dueño de la mascota a atender.

Para visualizar los datos de las citas debe darle click al cuadro ubicado debajo del botón de “**Actualizar Cita**” y encima de las opciones de la especie, luego debe darle click al nombre del dueño y enseguida, darle a Leer Cita.

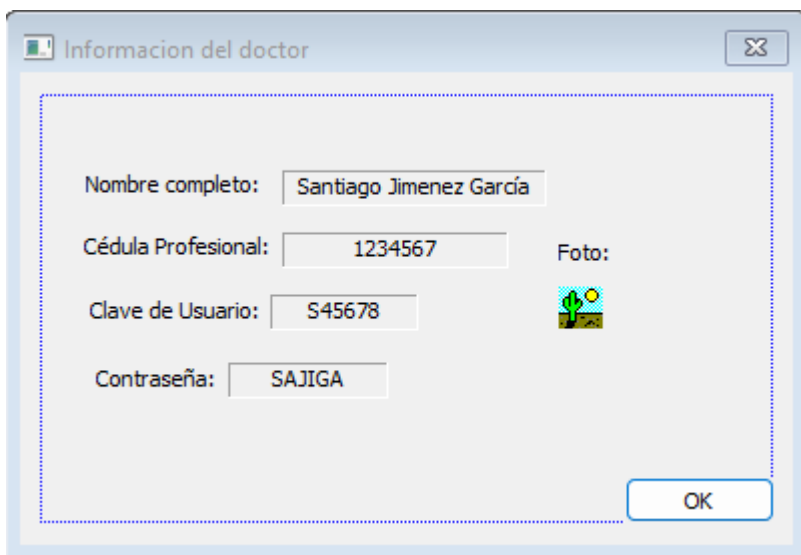
Esto, hará que los campos se llenen con la información del dueño, viendo así la información de dicha persona.

Además, para utilizar correctamente el botón de “Actualizar Lista”, primero deberá borrar los datos de los campos nuevamente e ingresar nuevos datos. Cuando ya los haya llenado, ahora si puede presionar el botón de “Actualizar Lista”.

Como último dato, tenemos “**Eliminar Cita**” el cual funciona en este orden: Seleccionar el nombre del dueño a eliminar, luego, presionar el botón de eliminar.

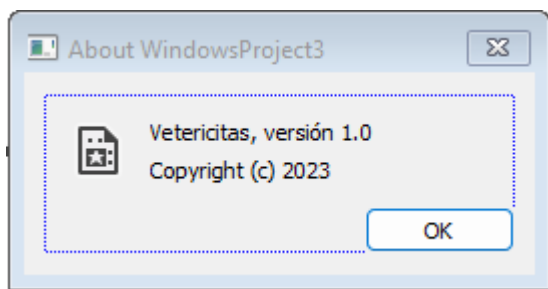
Información del doctor

Al seleccionar el apartado de “Información del doctor” ubicado en la sección de Help, le desplegará esta pantalla. En esta sección podrá visualizar solamente la información del usuario (doctor) que usará el programa.



About

En esta sección, podrá visualizar la información del programa.



Definición de #Listas y estructura de cada nodo.

Se utilizará una lista con sus respectivos nodos. Dichas estructuras serán de dos maneras. En la sección de ABC se utilizará una sola lista, la cual tendrá su estructura de dicha manera:

```
struct nodo_alumnos {  
    int matricula;  
    int costo;  
    char nombre[20];  
    char mascota[20];  
    char motivo[30];  
    char estatus[20];  
    char genero[1];  
    char foto[MAX_PATH];  
    int fechanac;  
    nodo_alumnos* ant;  
    nodo_alumnos* sig;  
};  
  
nodo_alumnos* aux = 0, * lista_alumnos = 0, * ultimo_alumno = 0,  
* nuevo_alumno = 0;
```

Esta estructura se utilizará para guardar los datos correctamente a mis listas doblemente ligadas, las cuales se utilizarán en los algoritmos siguientes.

ALGORITMO DE ALTA DE CITA

- 1- La información ingresada en el alta se guarda en las variables de la estructura, donde para mandar a llamar a AgregarAlumno, la cual toma como referencia una variable tipo puntero llamado nuevo para guardar correctamente todas las variables a mi nueva lista, el cual se le asignará definitivamente al final del llenado de datos. Todo esto realizado dentro de una función con sus respectivas validaciones.
- 2- La lista por utilizar se le asigna NULL, con el fin de que este sea el final de la lista.
- 3- El primer nodo (el cual es un puntero) apunta al inicio de la lista.
 - 3.1- El primer nodo se mueve al siguiente, siempre y cuando el puntero no sea NULL y que el siguiente nodo al que apunte el primer nodo tampoco sea NULL.
- 4- El puntero se agrega a un nuevo nodo, el cual guarda los datos ingresados
 - 4.1- El nuevo nodo que apunta al siguiente se iguala a NULL.
- 5- Insertamos el puntero nuevo a la lista
 - 5.1- Si la lista no está vacía, el nuevo nodo se agrega al final de la lista.
 - 5.1.1- Si la lista está vacía, esta apuntará al nuevo nodo.

Algoritmo Agenda

- 1- El nodo temporal se iguala a la lista.
- 2- La lista se recorre de inicio a fin, por medio de que temporal después de ser impreso, se iguale al siguiente dato de la lista, en este caso se utilizará el número de telefono. (La lista mostrará todos los datos incluidas en el #telefono).

Algoritmo Baja

- 1- Se busca una nueva variable tipo puntero para la baja llamada actual, la cual se iguala a la lista.
- 2- La lista buscará posición por posición hasta encontrar la variable, la cual usa como referencia el nombre del dueño.
- 3- Cuando se encuentre el nombre, se compara la dirección de memoria del nodo a eliminar contra la dirección en donde se encuentra el primer elemento de la lista.
 - 3.1- Si son iguales, el puntero "lista" se mueve al siguiente nodo.
 - 3.2- Si no, rompemos el vínculo que existe entre la lista y el nodo a eliminar.
- 4- Se libera la memoria asignada al nodo a eliminar.

Algoritmo Modificar

- 1- Se busca el nombre a modificar.
- 2- Solicitamos nuevamente los datos llamando a la función utilizada para dar de alta una cita.
- 3- La información ingresada en el alta se guarda en las variables de la estructura, donde para mandar a llamar a AgregarAlumno, la cual toma como referencia una variable tipo puntero llamado nuevo para guardar correctamente todas las variables a mi nueva lista, el cual se le asignará definitivamente al final del llenado de datos. Todo esto realizado dentro de una función con sus respectivas validaciones.
- 4- La lista por utilizar se le asigna NULL, con el fin de que este sea el final de la lista.
- 5- El primer nodo (el cual es un puntero) apunta al inicio de la lista.
 - 3.1- El primer nodo se mueve al siguiente, siempre y cuando el puntero no sea NULL y que el siguiente nodo al que apunte el primer nodo tampoco sea NULL.
- 6- El puntero se agrega a un nuevo nodo, el cual guarda los datos
 - 4.1- El nuevo nodo que apunta al siguiente se iguala a NULL.
- 7- Insertamos el puntero a la lista
 - 5.1- Si la lista no está vacía, el nuevo nodo se agrega al final de la lista.
 - 5.1.1- Si la lista está vacía, esta apuntará al nuevo nodo.