

**Matemáticas para videojuegos II**

**Docente: Luisa Alejandra Monreal Méndez**

**MANUAL DE USUARIO**

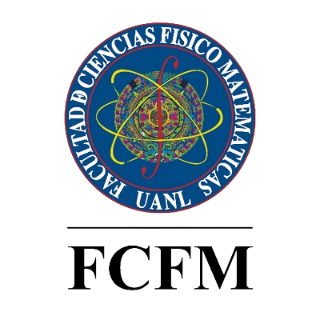
**Alumnos:**

* **Cristopher Ivan Ortiz Gonzalez**
* **Fernando Diaz Tamez**

**Grupo: 061**

**Fecha: 29 de noviembre del 2023**

**San Nicolás de los Garza, Nuevo León**



**Universidad Autónoma de Nuevo León**

**Facultad de Ciencias Fisico-Matemáticas**

**1.- Algoritmo general del programa.**

1. **Estructuras definidas:**
   * **Veterinario**: Almacena los datos de un veterinario, incluyendo su nombre, cédula, clave, contraseña y ruta de la foto.
   * **Citas**: Guarda información sobre las citas, como el nombre del cliente, teléfono, mascota, motivo de consulta, costo y fecha.
2. **Variables globales:**
   * **mascota** y **ao**: Matrices que almacenan nombres de mascotas y datos de citas, respectivamente.
   * Variables para almacenar datos de veterinarios (**nombreUsuario**, **cedula**, **clave**, **contraseña**, **path\_foto**) y para manipular listas de citas y veterinarios (**lista**, **Inicio**, **inicio**, etc.).
3. **Funciones importantes:**
   * **AgregarVeterinario**: Agrega un nuevo veterinario a la lista.
   * **Capturarveterinarios** y **capturarcitas**: Funciones relacionadas con la captura y almacenamiento de datos de veterinarios y citas, respectivamente.
   * **Datos\_Cargar** y **Datos\_Guardar**: Se encargan de cargar y guardar datos al iniciar o cerrar la aplicación.
   * Funciones de las ventanas (**Login**, **CrearCuenta**, **Menu**, etc.): Gestionan la interfaz gráfica, permitiendo iniciar sesión, crear cuentas, mostrar menús, altas, bajas, modificaciones y manejo de la agenda.
4. **El flujo general de la aplicación:**
   * El programa inicia con la función **WinMain**, carga datos de veterinarios y citas, crea la ventana de inicio de sesión (**Login**) y muestra la interfaz.
   * Desde **Login**, se puede iniciar sesión o crear una cuenta. Al iniciar sesión correctamente, se muestra el menú principal (**Menu**) del veterinario.
   * En el menú, se pueden realizar diversas acciones como altas, bajas, modificaciones, visualización de la agenda, entre otros.
   * El sistema permite la manipulación de datos de veterinarios y citas, así como el flujo de la interfaz a través de las distintas ventanas según las acciones del usuario.
5. **Gestión de ventanas:**
   * Cada ventana (**Login**, **CrearCuenta**, **Menu**, etc.) maneja sus propios eventos (**WM\_COMMAND**, **WM\_CLOSE**, etc.) y controla la lógica de la interfaz gráfica.
6. **Algoritmo para la función Alta:**
7. **Mensaje WM\_INITDIALOG**:
   * Obtiene la hora local del sistema.
   * Obtiene el texto del **DateTimePicker** y lo establece en el control **IDC\_StaticFecha**.
   * Busca un veterinario en una lista enlazada utilizando un nombre de usuario.
   * Si encuentra el veterinario:
     + Establece el nombre del veterinario en el control **IDC\_MEDICONOMBRE**.
     + Carga la imagen del veterinario en el control **IDC\_MEDICOIMAGENALTA**.
   * Agrega elementos al ComboBox **ID\_MASCOTAS**.
   * Obtiene el handle del botón **ID\_BTN\_CAPTURAR** y lo deshabilita.
8. **Mensaje WM\_COMMAND**:
   * **Case ID\_BTN\_AGREGARMASCOTA**:
     + Obtiene el texto ingresado en un cuadro de edición.
     + Agrega ese texto al ComboBox **ID\_MASCOTAS**.
     + Limpia el cuadro de edición después de agregar el texto.
   * **Case ID\_BTN\_CAPTURAR**:
     + Obtiene la fecha seleccionada.
     + Verifica si la mascota seleccionada está en la lista de opciones del ComboBox **ID\_MASCOTAS**.
     + Si la fecha seleccionada es futura:
       - Crea una nueva cita con la información ingresada en los controles.
       - Ordena la cita por fecha.
       - Muestra un mensaje de confirmación y limpia los controles.
     + Si la fecha seleccionada es pasada:
       - Muestra un mensaje de advertencia y establece la fecha actual en el **DateTimePicker**.
   * **Case ID\_SALIRALTA**:
     + Pide confirmación para salir y limpia los controles si se confirma.
9. **Mensajes WM\_CLOSE y WM\_DESTROY**:
   * Procesa el cierre de la ventana.
10. **Algoritmo para la función Baja:**
11. **Mensaje WM\_INITDIALOG**:
    * Busca un veterinario en una lista enlazada utilizando un nombre de usuario.
    * Si encuentra el veterinario:
      + Establece el nombre del veterinario en el control **IDC\_MEDICONOMBRE**.
      + Carga la imagen del veterinario en el control **IDC\_MEDICOIMAGENBAJA**.
12. **Mensaje WM\_COMMAND**:
    * **Case BTN\_SELECCIONAR\_BAJA**:
      + Busca una cita por el nombre del cliente ingresado.
      + Si encuentra la cita:
        - Muestra un mensaje de éxito.
        - Muestra la información de la cita en diferentes controles.
      + Si no encuentra la cita:
        - Muestra un mensaje de error.
    * **Case BTN\_SALIR\_BAJA**:
      + Pide confirmación para salir y limpia los controles si se confirma.
    * **Case BTN\_ELIMINAR**:
      + Elimina una cita según el nombre del cliente ingresado.
      + Muestra un mensaje de confirmación si se elimina con éxito.
      + Si no encuentra la cita, muestra un mensaje de información.
13. **Mensajes WM\_CLOSE y WM\_DESTROY**:
    * Procesa el cierre de la ventana.
14. **Función Modificar**
15. **Inicialización de la interfaz y carga de datos iniciales:**
    * Busca un veterinario en base al nombre de usuario.
    * Si encuentra al veterinario, muestra su información en la interfaz (nombre, imagen, etc.).
    * Agrega elementos a un combo box (**CB\_MODTIPO**) con ciertos datos.
    * Deshabilita un campo de fecha.
16. **Manejo de eventos de la interfaz:**
    * Detecta eventos como clics de botón (**BTN\_SELECTMOD**, **BTN\_MODIFICAR**, **BTN\_SALIRMODIFICAR**) o cambios en los controles.
    * Realiza validaciones y acciones según el evento, como la modificación de datos de citas, selección de elementos, etc.
17. **Función Agenda**
18. **Inicialización de la interfaz y carga de datos iniciales:**
    * Similar a la función **Modificar**, busca y muestra información del veterinario en la interfaz.
    * Ordena las citas por fecha y las agrega a un list box (**IDC\_LISTAMOSTRAR**) en la interfaz.
19. **Manejo de eventos de la interfaz:**
    * Detecta eventos como clics de botón (**BTN\_ELIMINARCITA**, **BTN\_MODIFICARCITA**, **BTN\_AGREGARCITA**, **BTN\_MOSTRARDATOS**, **BTN\_SALIRAGENDA**) o cambios en los controles.
    * Realiza acciones como mostrar detalles de citas, agregar nuevas citas, eliminar citas, etc.
20. **Función AgregarVeterinario**

Esta función parece ser una parte de la lógica para agregar un nuevo veterinario al sistema, pero no está integrada en las funciones anteriores. Toma ciertos datos (nombre, cédula, usuario, contraseña, ruta de la foto) y crea un nuevo nodo de tipo **Veterinario** en una estructura enlazada.

1. **Función Capturarveterinarios**
2. **Abre un cuadro de diálogo para seleccionar la ruta y nombre del archivo donde se guardarán los datos de veterinarios en formato binario (\*.bin o \*.txt).**
3. **Guarda los datos de los veterinarios en el archivo binario seleccionado:**
   * Abre el archivo para escritura.
   * Recorre la lista de veterinarios y escribe sus datos en el archivo, separando cada dato con un salto de línea.
   * Cierra el archivo y muestra un mensaje de éxito o error.
4. **Función capturarcitas**
5. **Abre un cuadro de diálogo para seleccionar la ruta y nombre del archivo donde se guardarán los datos de citas en formato binario (\*.bin o \*.txt).**
6. **Guarda los datos de las citas en el archivo binario seleccionado:**
   * Abre el archivo para escritura.
   * Recorre la lista de citas y escribe sus datos en el archivo, incluyendo la fecha formateada, separando cada dato con un salto de línea.
   * Cierra el archivo y muestra un mensaje de éxito o error.
7. **Función Datos\_Cargar**
8. **Abre un archivo binario predefinido para cargar datos de citas:**
   * Lee datos de citas desde el archivo binario y los carga en la lista enlazada de citas.
9. **Función Datos\_Guardar**
10. **Abre un archivo binario predefinido para guardar datos de citas:**
    * Guarda los datos de citas de la lista enlazada en el archivo binario.
11. **Función Datos\_Veterinario\_Cargar**
12. **Abre un archivo binario predefinido para cargar datos de veterinarios:**
    * Lee datos de veterinarios desde el archivo binario y los carga en la lista enlazada de veterinarios.
13. **Función Datos\_Veterinario\_Guardar**
14. **Abre un archivo binario predefinido para guardar datos de veterinarios:**
    * Guarda los datos de veterinarios de la lista enlazada en el archivo binario.
15. **Función existeCitaEnFecha**
16. **Verifica si ya existe una cita para la fecha dada:**
    * Recorre la lista de citas y compara la fecha proporcionada con las fechas de las citas existentes.
    * Devuelve **true** si encuentra una cita para esa fecha, de lo contrario, devuelve **false**.