

Trabajo Práctico: Archivos



**Carrera**: Tecnicatura Superior en Desarrollo de Software

Espacio curricular: Programación 1

Campo de formación: General

**Docente**: Daniela Paz Nacusse

**Alumno**: Rodriguez Enzo Lautaro

Fecha de presentación:22/10/2024

# Consigna

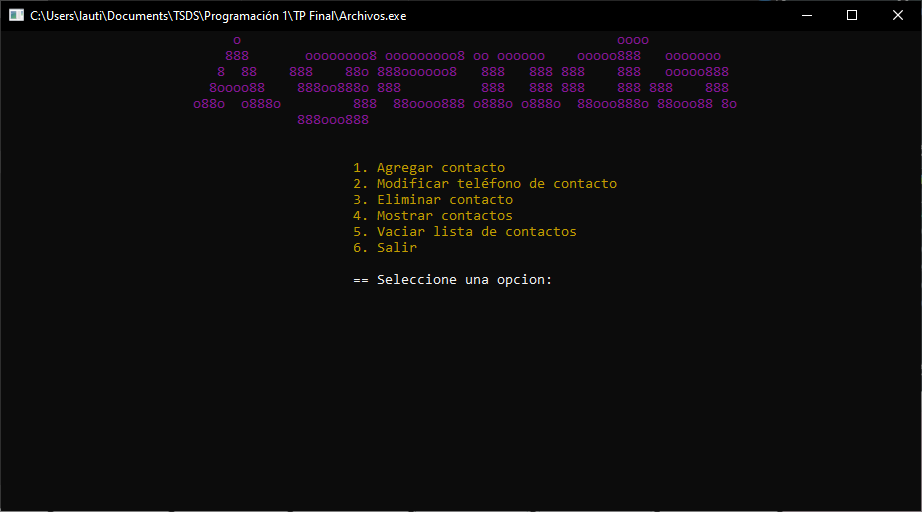
1. Desarrollar un ejemplo usando archivos y todos sus tratamientos (de texto).
2. Definir la situación problemática a resolver con el programa a desarrollar.

# Resolución

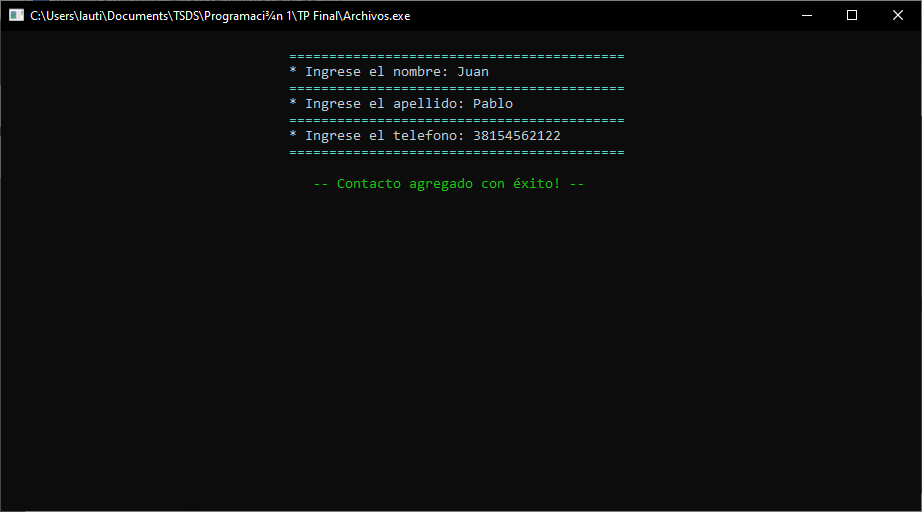
Para cumplir con las consignas desarrollé un programa que solucione la siguiente situación problemática:  
Una empresa indie de videojuegos quiere una forma de poder agendar a los empleados sin utilizar la aplicación del celular. Además, el programa que necesitan debe guardar el nombre, apellido y teléfono de cada empleado además de asignarle un ID único. Este programa también debe contar con la posibilidad de agregar y eliminar contactos, mostrar la lista de contactos, y una opción para vaciar toda la lista.

# Ejecución del programa

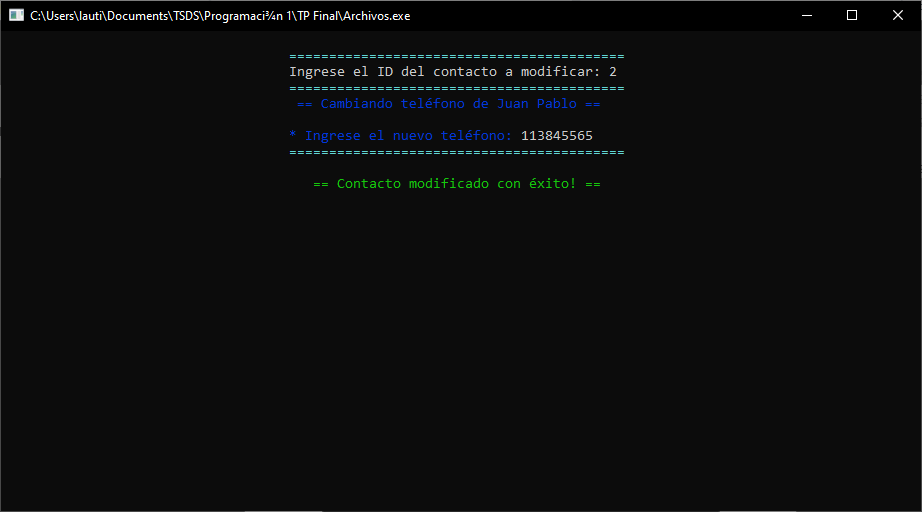
Al ingresar al programa, lo primero que se puede observar es el menú del mismo, el cual cuenta con diferentes opciones.



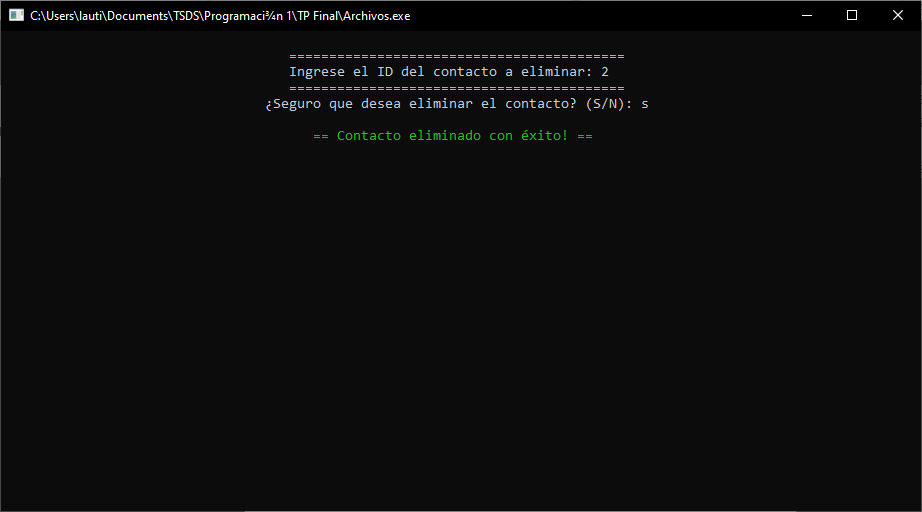
Al ingresar en la opción 1 (Agregar contacto), el programa nos irá pidiendo datos de manera secuencial y al presionar una tecla volverá al menú.



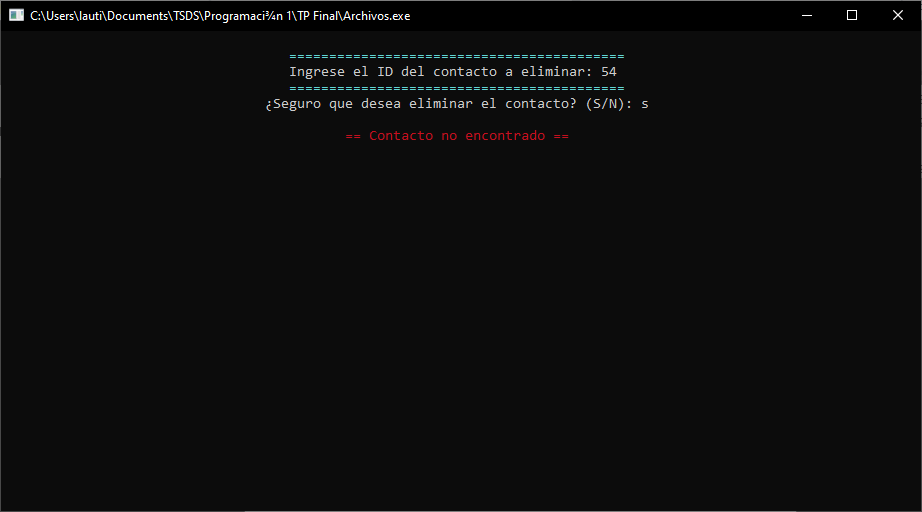
En la opción 2 podremos modificar el número de algún contacto mediante el ID, el cual nos pedirá al ingresar en la opción.



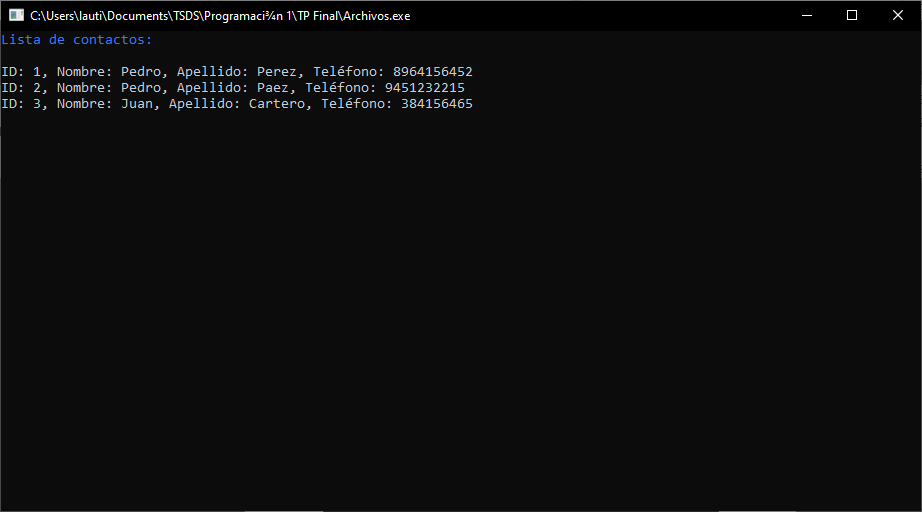
En la opción 3, podemos eliminar un contacto de la agenda mediante su ID



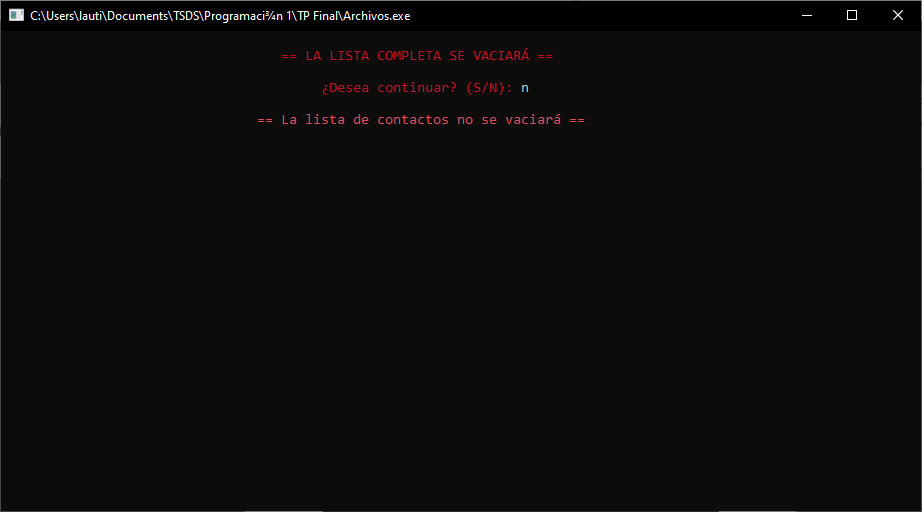
En caso de ingresar un ID que no exista mostrará lo siguiente



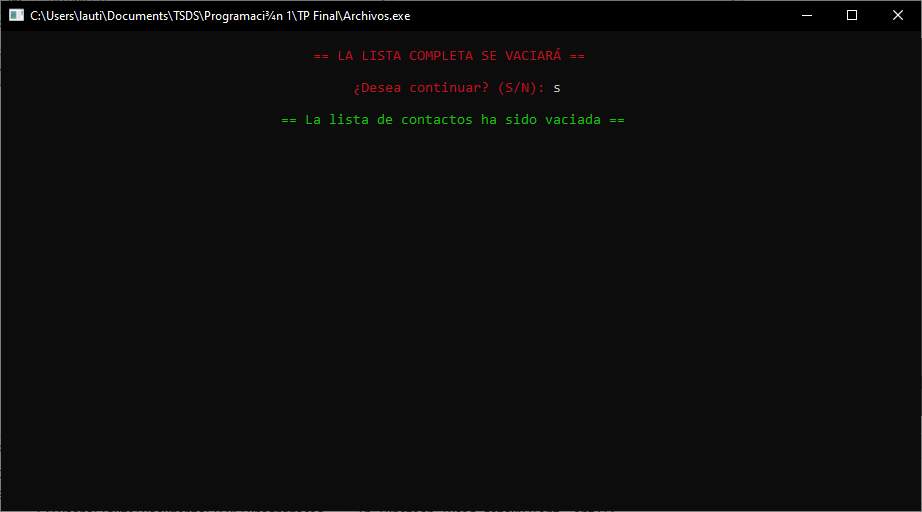
La opción 4 nos permite ver la agenda completa, en la que se muestra toda la información de los contactos.



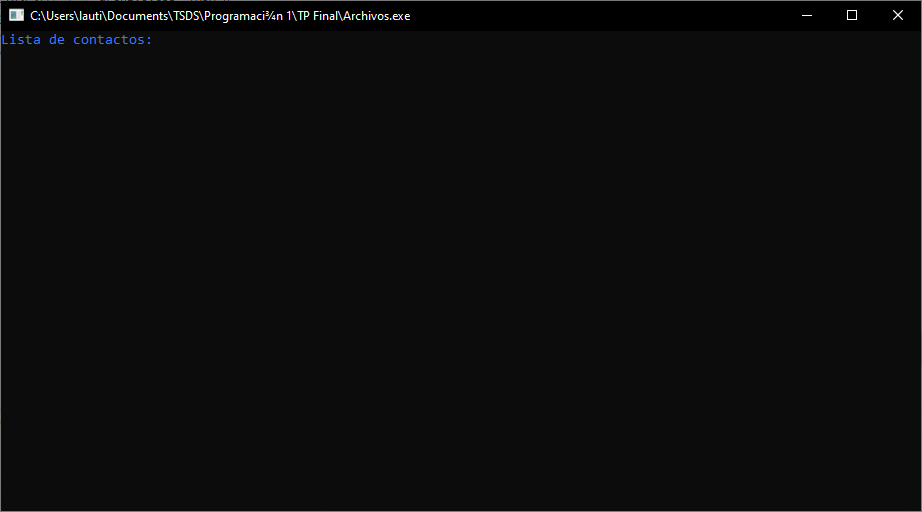
La opción 5 es para vaciar la lista de contactos completa, al usuario se le realizará una pregunta para que confirme si desea o no vaciar la lista, en caso de negarse mostrará lo siguiente



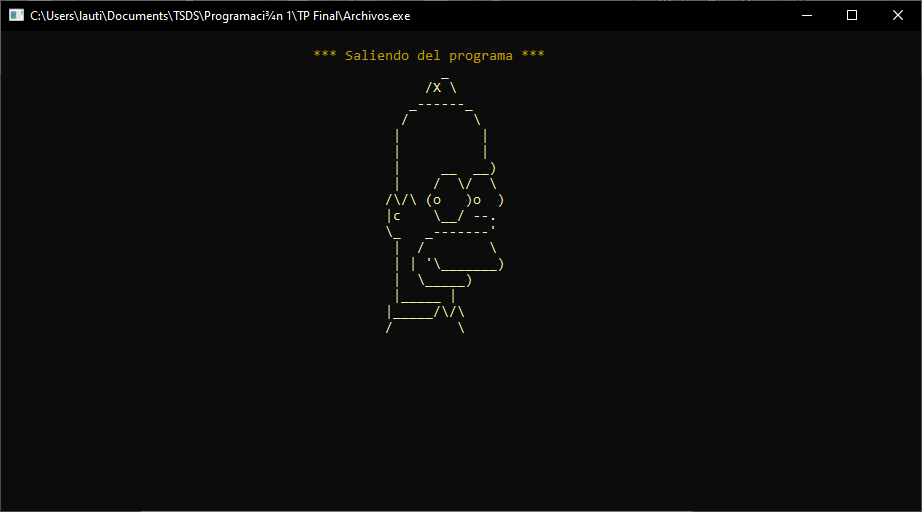
Y en caso de aceptar:



Vuelvo a ingresar a la opción 4 (mostrar lista) para comprobar que esté vacía

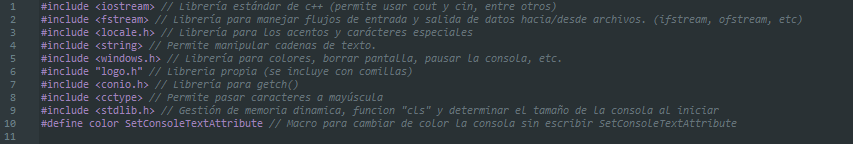


Y, por último, al seleccionar la opción 6 del menú (Salir) mostrará lo siguiente y al presionar cualquier tecla el programa se dejará de ejecutar.



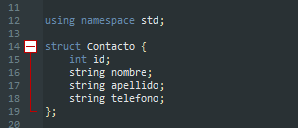
# Explicación del código

## Librerías utilizadas

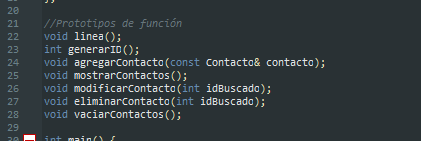


## Código

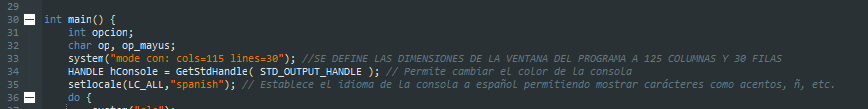
Esta estructura define cómo se almacenan los contactos en el programa, con cuatro campos: id (un entero que sirve como identificador único), nombre, apellido, y teléfono.



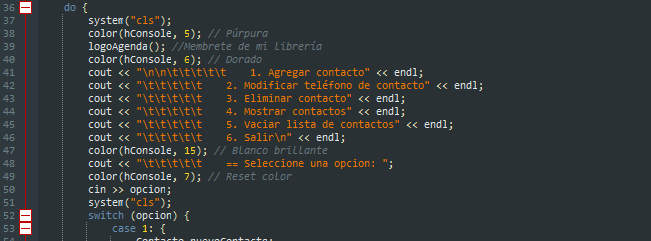
Estos son los prototipos de función (sirven para hacerle saber al compilador que la función existe)



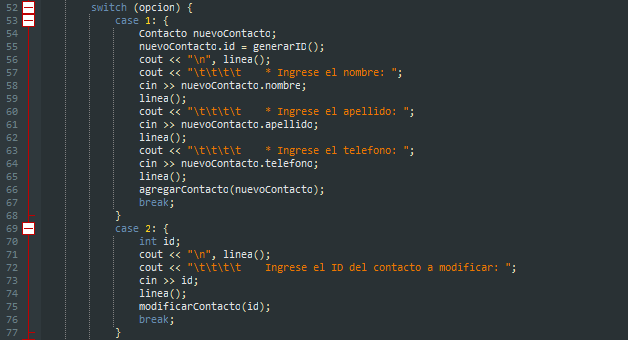
En la función main no hay mucho que explicar, comienza declarando las variables que se van a utilizar, definiendo el tamaño de la consola y el setlocale para poder utilizar acentos y caracteres especiales.



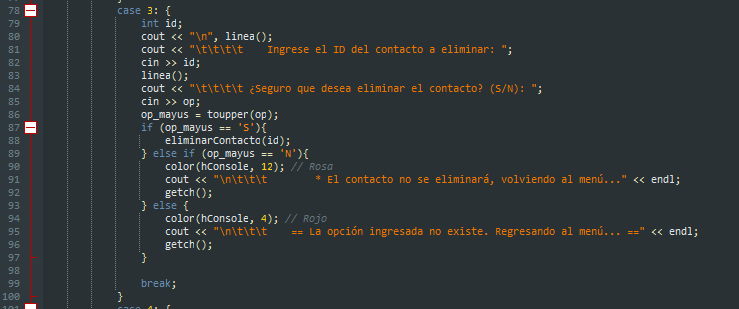
A continuación, hay un do-while para mostrar el menú y guardar la opción que el usuario elija en la variable “opción”.



Luego, esa variable “opción” nos permite direccionar al usuario mediante un switch. En caso de que el usuario ingrese la opción 1 se crea un nuevo contacto y se le asigna un nuevo id (mediante la función generarID), luego se le piden los datos al usaurio y se los van guardando en la estructura del nuevo contacto, y por ultimo se ejecuta la función agregarContacto para agregar el nuevo contacto al archivo.

En caso de ingresar la opción 2, se le solicita al usuario que ingrese el ID que quiera modificar y lo modifica mediante la función modificarContacto (todas las funciones se explicarán mas adelante)  


Al ingresar la opción 3, se le solicita al usuario que ingrese el ID del contacto que quiera eliminar y luego se hace una confirmación para eliminar dicho contacto, en caso de que el usuario ingrese la opción “s”, esta se pasa a mayúsculas con la función toupper() y entra a la función eliminarContacto, en caso de ingresar “n” lo devuelve al menú y si ingresa una opción que no sean n o s, le avisa que esa opción no existe y lo devuelve al menú.

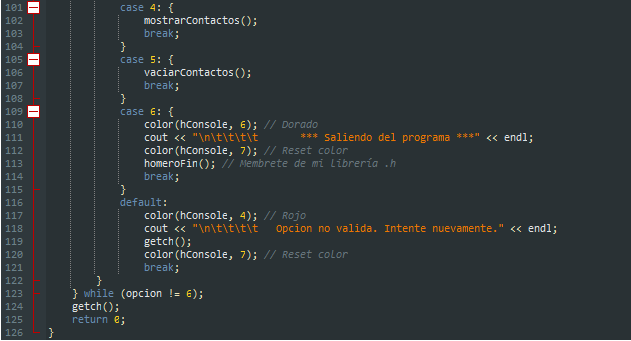


En el caso 4 simplemente ejecuta la función mostrarContactos.

En el caso 5 lo mismo, pero con la función vaciarContactos

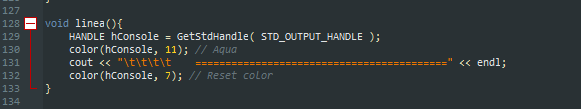
El caso 6 muestra un membrete de mi librería .h y la frase “saliendo del programa”

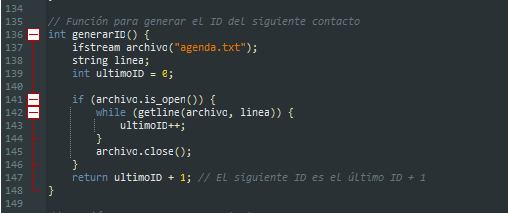
Por último, el default, que simplemente le avisa al usuario que la opción ingresada es incorrecta. El programa finaliza al ingresar la opción 6.



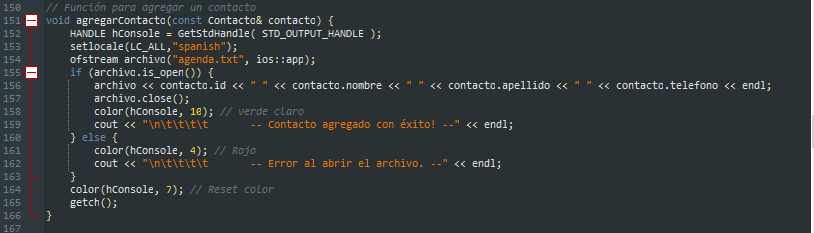
## Funciones explicadas

La función “linea()” lo único que hace es mostrar una linea de formada por ‘=’ de color aqua.

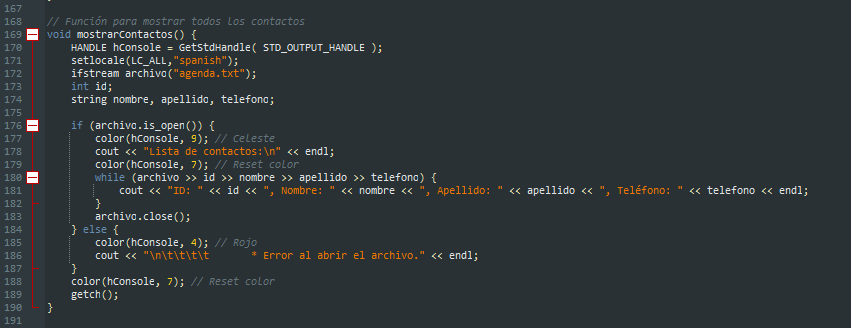


Esta función cuenta el número de líneas en el archivo (cada línea representa un contacto) y luego devuelve un nuevo ID que es una unidad mayor que el número de contactos existentes. Esto asegura que cada nuevo contacto tenga un ID único y consecutivo. 

La siguiente función abre el archivo con “ofstream” que es una función de la librería “fstream”, la cual revisa que el archivo “agenda.txt” exista y lo abre para modificar, en caso contrario lo crea. Una vez abierto el archivo agrega el nuevo contacto del cual se recopilaron los datos en el switch y por último muestra que el contacto fue agregado con éxito.



La función “mostrarContactos()” lee todos los contactos almacenados en el archivo agenda.txt y los muestra en la consola, con un formato claro y organizado (ID, nombre, apellido, teléfono). Si el archivo no existe o no puede ser abierto, muestra un mensaje de error. También utiliza colores para mejorar la presentación en la consola.



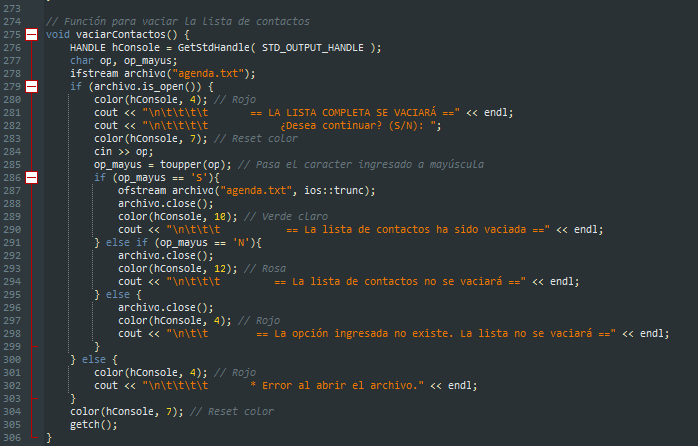
La función “modificarContacto()” busca un contacto en el archivo agenda.txt por su ID, le permite al usuario cambiar el número de teléfono de ese contacto, y luego guarda todos los contactos (modificados o no) en un archivo temporal. Después de esto, el archivo original se reemplaza con el archivo temporal, asegurando que los cambios se guarden correctamente. Si no se encuentra el contacto, se muestra un mensaje de error.



La función “eliminarContacto()” busca un contacto en el archivo agenda.txt según su ID. Si lo encuentra, lo elimina al no copiar dicho contacto al archivo temporal. Luego, el archivo original se elimina y es reemplazado por el temporal, que contiene todos los contactos menos el que se eliminó. Si el contacto no se encuentra, se muestra un mensaje de error.



La función “vaciarContactos()” permite borrar todos los contactos de la agenda, solicitando previamente una confirmación del usuario. Si el usuario confirma, el archivo agenda.txt se vacía usando el modo truncado. Si el usuario cancela o ingresa una opción inválida, no se hace ningún cambio. Si el archivo no se puede abrir, se muestra un mensaje de error.



# Conclusión

El manejo de archivos en programación es fundamental ya que permite a los desarrolladores almacenar, recuperar y manipular datos de manera eficiente, ya que permite