

MEMORIA TÉCNICA PROYECTO **GRUPAL FUNDAMENTOS DE** **PROGRAMACIÓN**

Francho Ladaga.

Javier Muzás.

David Villaroya.

(14.01.2024

Índice:

- 1) Resumen. (3)**
- 2) Una introducción al documento en detalle.**
- 3) Decisiones de implementación.**
- 4) Una sección de resultados que muestre imágenes del programa funcionando.**
- 5) Una pequeña sección de conclusiones extraídas del proyecto.**

1) Resumen de lo que se ha hecho en el proyecto, con una pequeña conclusión.

En este proyecto, se desarrolló un programa de gestión de cuentas cifradas en C. El programa permite a los usuarios autenticarse, ver, agregar y eliminar cuentas almacenadas. Se implementó un sistema de archivos para almacenar la información de las cuentas de manera segura. La autenticación se realiza mediante una contraseña maestra, y las contraseñas no se almacenan directamente en el programa. La interfaz incluye un menú de opciones para el usuario. Se tomaron decisiones de implementación para garantizar la robustez y seguridad del programa. A pesar de que se asumen funciones no proporcionadas, como la lectura de datos y el espacio de memoria, se presenta una estructura clara y concisa para futuras extensiones del proyecto.

En la etapa de desarrollo, se implementó una robusta autenticación mediante contraseñas duras y se estableció un sistema de archivos que crea y manipula archivos correspondientes a cada usuario. Se adoptó una estructura modular para las funciones principales, permitiendo mayor flexibilidad y facilitando futuras actualizaciones del programa.

El programa guía a sus nuevos usuarios con instrucciones claras y da opciones de gestión de cuentas a los usuarios existentes, manteniendo la seguridad como un principio clave.

En resumen, el proyecto logra desarrollar un sistema eficiente y seguro para la gestión cifrada de cuentas en C, sentando las bases para futuras expansiones y mejoras.

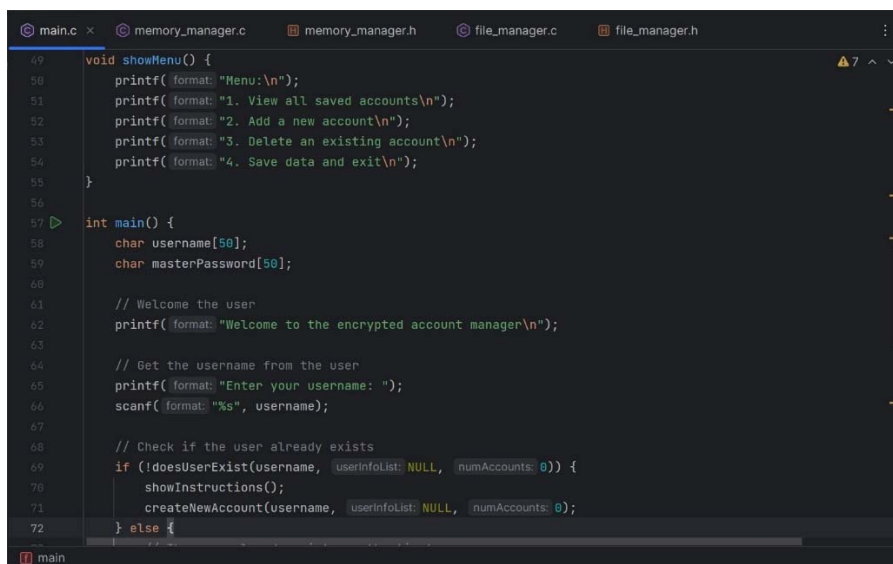
2) Una introducción al documento en detalle.

La memoria técnica documenta el desarrollo de un programa de gestión de cuentas cifradas en C. Se presenta un resumen del proyecto, decisiones de implementación, resultados visuales y conclusiones. La importancia del trabajo radica en ofrecer a los usuarios una herramienta segura y eficiente para gestionar sus cuentas de manera cifrada. La aplicación se centra en la autenticación, visualización, agregación y eliminación de cuentas, implementando un sistema de archivos para almacenar la información de manera segura.

3) Decisiones de implementación.

En este proyecto se han tomado decisiones clave para su diseño, como el uso de funciones modulares, un sistema de archivos para almacenar datos y la implementación de la autenticación mediante una contraseña maestra. Se utilizó una estructura modular para facilitar futuras expansiones y mantenimiento del código.

4) Resultados



```
49 void showMenu() {
50     printf( format: "Menu:\n");
51     printf( format: "1. View all saved accounts\n");
52     printf( format: "2. Add a new account\n");
53     printf( format: "3. Delete an existing account\n");
54     printf( format: "4. Save data and exit\n");
55 }
56
57 int main() {
58     char username[50];
59     char masterPassword[50];
60
61     // Welcome the user
62     printf( format: "Welcome to the encrypted account manager\n");
63
64     // Get the username from the user
65     printf( format: "Enter your username: ");
66     scanf( format: "%s", username);
67
68     // Check if the user already exists
69     if (!doesUserExist(username, userInfoList: NULL, numAccounts: 0)) {
70         showInstructions();
71         createNewAccount(username, userInfoList: NULL, numAccounts: 0);
72     } else {
```

```
main.c  memory_manager.c  memory_manager.h  file_manager.c  file_manager.h
25 }
26
27 void fillUserInfo(UserInfo *userInfo) {
28     // Prompts the user to enter the username and password
29     printf( format: "Ingrese el nombre de usuario: ");
30     userInfo->username = (char *)malloc( Size: 50 * sizeof(char)); // Reserva memoria para el nombre de usuario
31     scanf( format: "%s", userInfo->username);
32
33     printf( format: "Ingrese la contraseña: ");
34     userInfo->password = (char *)malloc( Size: 50 * sizeof(char)); // Reserva memoria para la contraseña
35     scanf( format: "%s", userInfo->password);
36
37     // Store the length of the strings
38     userInfo->usernameLength = strlen( Str: userInfo->username);
39     userInfo->passwordLength = strlen( Str: userInfo->password);
40
41     // Frees previous memory, if it has already been allocated
42     free( Memory: userInfo->username);
43     free( Memory: userInfo->password);
44
45     // Reserve memory for character strings
46     userInfo->username = (char *)malloc( Size: (userInfo->usernameLength + 1) * sizeof(char));
47     userInfo->password = (char *)malloc( Size: (userInfo->passwordLength + 1) * sizeof(char));
48 }
```

```
main.c  memory_manager.c  memory_manager.h  file_manager.c  file_manager.h
1 > #include <stdio.h>
2
3
4
5
6 static int calculateChecksum(const char *data, int length, const char *key) {
7     int checksum = 0;
8     for (int i = 0; i < length; ++i) {
9         checksum += (int)data[i];
10    }
11    return checksum;
12 }
13
14 static void xorEncryptDecrypt(char *data, int length, const char *key) {
15     int keyLength = strlen( Str: key);
16     for (int i = 0; i < length; ++i) {
17         data[i] = data[i] ^ key[i % keyLength];
18     }
19 }
20
21 int readUserInfo(const char *path, const char *encryptionKey, UserInfo **userInfoList, int *numAccounts) {
22     // Implement reading binary file and decrypting with the encryption key
23     // Return the list of accounts and the number of accounts read from the file
24 }
25
26 int writeUserInfo(const char *path, const UserInfo *userInfoList, int numAccounts, const char *encryptionKey) {
27     // Implement writing binary file and encrypting with the encryption key
28 }
```

5) Conclusiones

Para finalizar, se extraen lecciones sobre la importancia de la modularidad, la seguridad en la gestión de contraseñas y la necesidad de manejar errores del usuario. Se sugieren mejoras para otras próximas versiones y se destacan las dificultades superadas durante el desarrollo que hemos hecho durante el proyecto. La memoria técnica se encuentra disponible en el repositorio de Github.
