

Sistema De Gestión De Estudiantes

Nombre Completo del Estudiante: Wayra Tiziana Sandoval Pernía

Nombre de la Institución: Técnica N°2

Curso: Laboratorio de Programación 5° 2° B

Profesor: Mansilla Muñoz York E.

Fecha: 16/09/2025

Objetivos del proyecto:

Los objetivos de este proyecto son básicamente desarrollar un programa en lenguaje c que permita gestionar estudiantes usando archivos binarios (de ahí su nombre: sistema de gestión de estudiantes), Este sistema nos debería Permitir Registrar un estudiante, Buscar por legajo, Listar estudiantes, Modificar y Baja lógica.

Herramientas utilizadas:

Para este proyecto las herramientas que se usaron fueron Visual Studio Code, GCC (MinGW), Compilador c, Word (para hacer el documento luego al guardarlo lo cambio a pdf), GitHub.

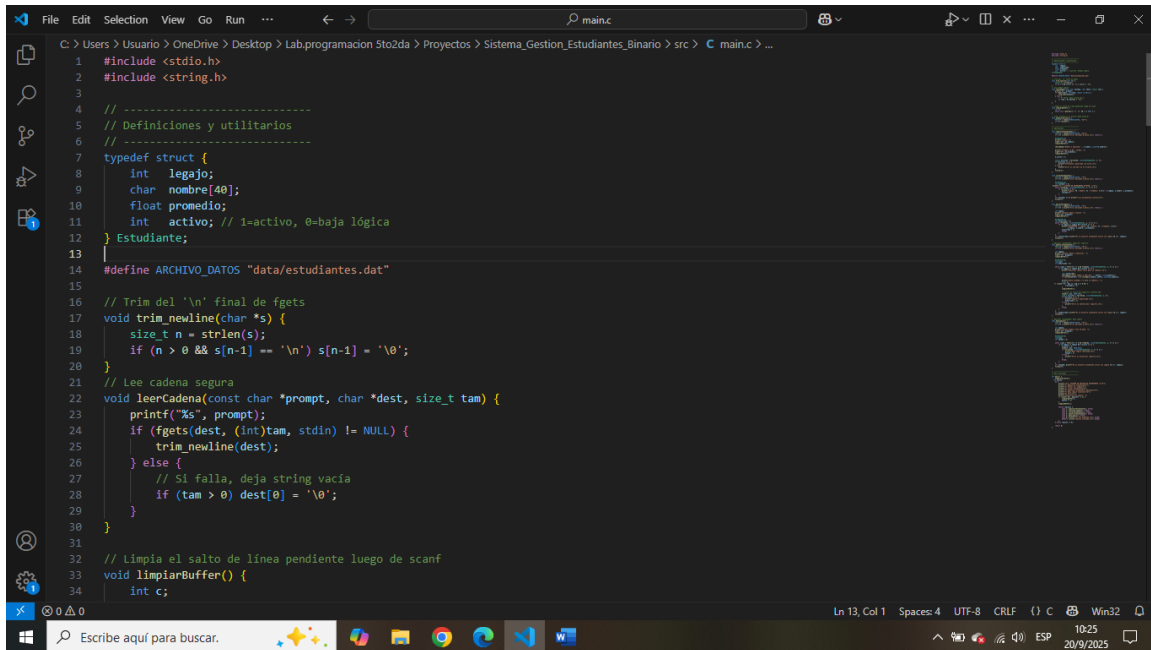
Descripción del código:

Este código es un sistema de gestión de estudiantes que con un menú de opciones en las cuales esta Registrar un estudiante, Buscar por legajo, Listar estudiantes, Modificar y Baja lógica.

Las cuales nos ayudan a ejecutar este sistema, Además de que todo esto es mediante el uso de archivos binarios.

Mas abajo voy a proceder a explicar un paso a paso de este con imágenes.

Definiciones y utilitarios:

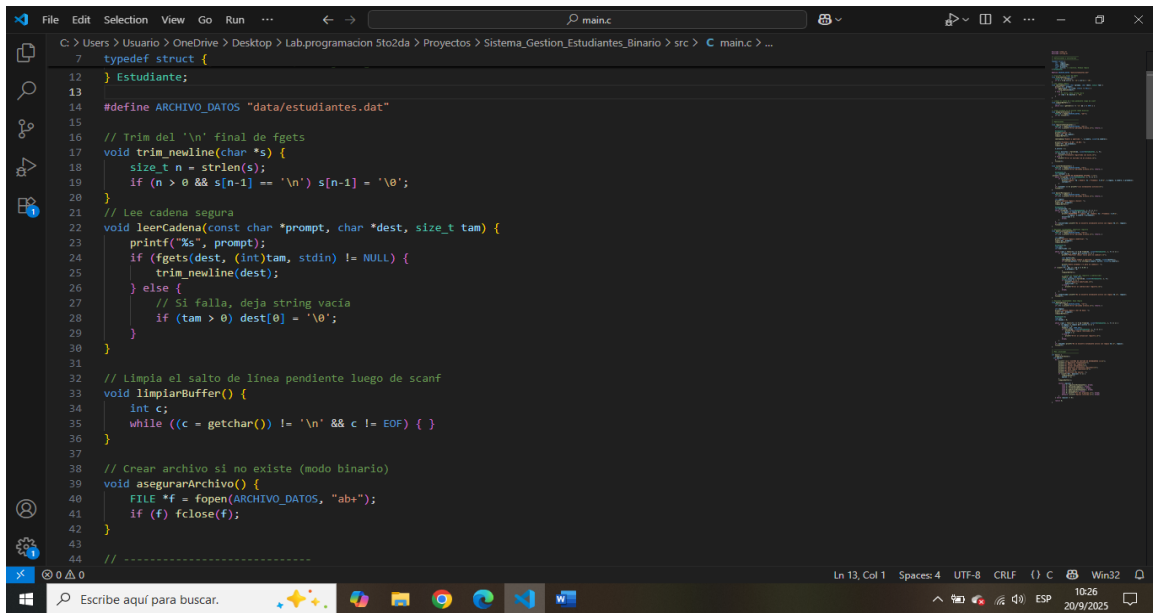


```
1 #include <stdio.h>
2 #include <string.h>
3
4 // -----
5 // Definiciones y utilitarios
6 // -----
7 typedef struct {
8     int legajo;
9     char nombre[40];
10    float promedio;
11    int activo; // 1=activo, 0=baja lógica
12 } Estudiante;
13
14 #define ARCHIVO_DATOS "data/estudiantes.dat"
15
16 // Trim del '\n' final de fgets
17 void trim_newline(char *s) {
18     size_t n = strlen(s);
19     if (n > 0 && s[n-1] == '\n') s[n-1] = '\0';
20 }
21
22 // Lee cadena segura
23 void leerCadena(const char *prompt, char *dest, size_t tam) {
24     printf("%s", prompt);
25     if (fgets(dest, (int)tam, stdin) != NULL) {
26         trim_newline(dest);
27     } else {
28         // Si falla, deja string vacía
29         if (tam > 0) dest[0] = '\0';
30     }
31 }
32
33 // Limpia el salto de línea pendiente luego de scanf
34 void limpiarBuffer() {
35     int c;
```

En esta imagen se pueden observar varias cosas como por ejemplo hay una estructura del 7 al 12 que nos va a estar definiendo el tipo de variables que están dentro del dato llamado estudiante, con comandos como por ejemplo int, char, float para definir el tipo de variable con el que se va a llenar lo que necesitamos que es el legajo, el nombre y el promedio.

Después de el 17 al 20 tenemos lo que hace que se elimine el salto de línea (`\n`), esto es más que nada para evitar que el nombre de los estudiantes no queden con un salto de línea Extra. Además de que el if que esta nos ayuda a saber si la cadena esta vacía y también a ver si el ultimo carácter es un salto de línea

Luego en el numero 22 tenemos lo que nos ayuda a leer cadenas seguras evitando el desbordamiento de buffer, Después tenemos la condicional if, else la cual va a tratar de leer el texto para luego guardarlo en dest, si llegara a funcionar nos va a devolver algo distinto a "null", pero si llegara a fallar si nos va a devolver null, Si todo va bien y se puede leer todo bien va a limpiar el salto de línea, pero si falla y no se puede leer va a dejar la cadena vacía.

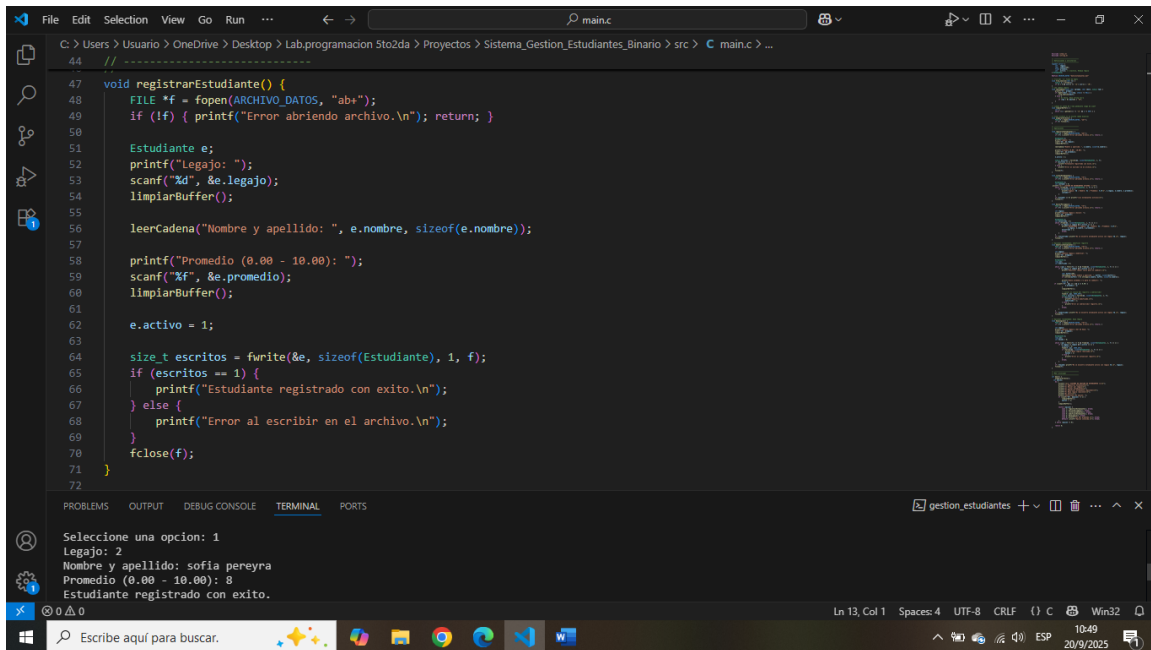


```
7 typedef struct {
12 } Estudiante;
13
14 #define ARCHIVO_DATOS "data/estudiantes.dat"
15
16 // Trim del '\n' final de fgets
17 void trim_newline(char *s) {
18     size_t n = strlen(s);
19     if (n > 0 && s[n-1] == '\n') s[n-1] = '\0';
20 }
21
22 // Lee cadena segura
23 void leerCadena(const char *prompt, char *dest, size_t tam) {
24     printf("%s", prompt);
25     if (fgets(dest, (int)tam, stdin) != NULL) {
26         trim_newline(dest);
27     } else {
28         // Si falla, deja string vacia
29         if (tam > 0) dest[0] = '\0';
30     }
31
32 // Limpia el salto de línea pendiente luego de scanf
33 void limpiarBuffer() {
34     int c;
35     while ((c = getchar()) != '\n' && c != EOF) { }
36 }
37
38 // Crear archivo si no existe (modo binario)
39 void asegurarArchivo() {
40     FILE *f = fopen(ARCHIVO_DATOS, "ab+");
41     if (f) fclose(f);
42 }
43
44 // -----
45
```

de esta imagen solo vamos a seguir del numero 32 para abajo ya que lo anterior esta explicado anteriormente (ósea en este mismo archivo), desde el numero 33 al 35 lo que va a hacer es básicamente limpiar el buffer para evitar que queden caracteres sin leer.

Por último, del 39 al 41 esto básicamente sirve para que el programa tenga siempre un archivo en el cual trabajar, por q digo esto, es por que lo que hace es abrir el archivo en modo ab+ y si este no llegara a existir crea uno.

Registrar estudiante:



```
44 //
45
46 void registrarEstudiante() {
47     FILE *f = fopen(ARCHIVO_DATOS, "ab+");
48     if (!f) { printf("Error abriendo archivo.\n"); return; }
49
50     Estudiante e;
51     printf("Legajo: ");
52     scanf("%d", &e.legajo);
53     limpiarBuffer();
54
55     leerCadena("Nombre y apellido: ", e.nombre, sizeof(e.nombre));
56
57     printf("Promedio (0.00 - 10.00): ");
58     scanf("%f", &e.promedio);
59     limpiarBuffer();
60
61     e.activo = 1;
62
63     size_t escritos = fwrite(&e, sizeof(Estudiante), 1, f);
64     if (escritos == 1) {
65         printf("Estudiante registrado con exito.\n");
66     } else {
67         printf("Error al escribir en el archivo.\n");
68     }
69     fclose(f);
70 }
71
72
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

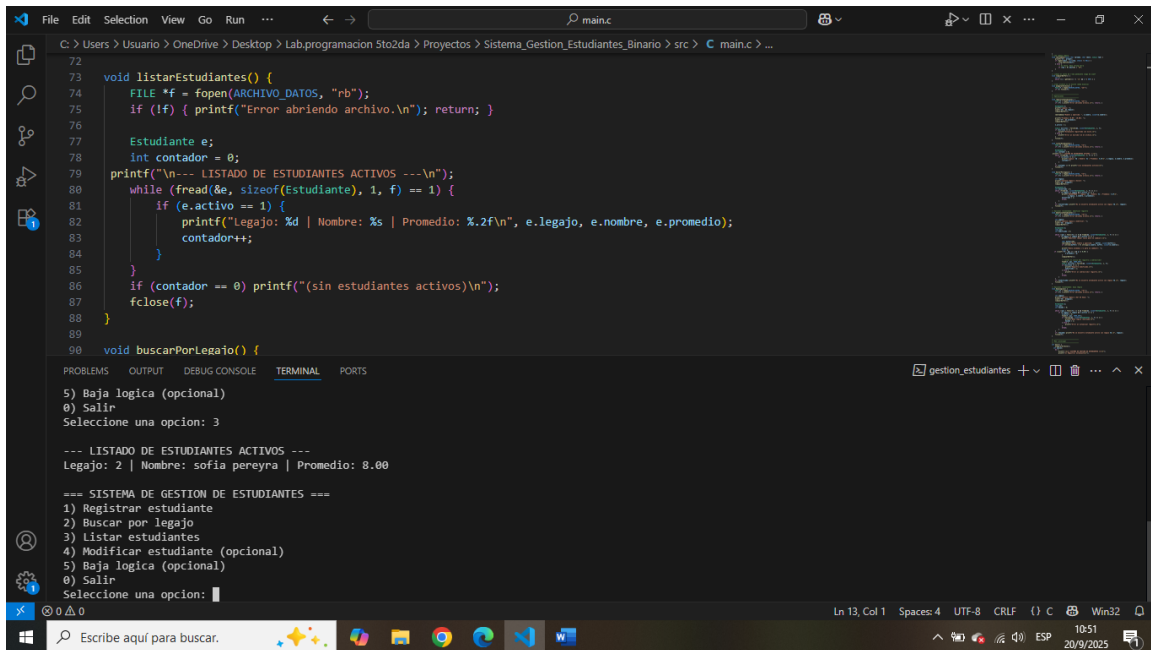
gestion_estudiantes + ▾

Seleccione una opcion: 1
Legajo: 2
Nombre y apellido: sofia pereyra
Promedio (0.00 - 10.00): 8
Estudiante registrado con exito.

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32 10:49 20/9/2025

Esta es la opción 1 de nuestro menú de opciones (Que se va a explicar mas abajo en otra imagen sobre el menú) esta opción es registrar un estudiante, básicamente a lo largo de este código lo que va a hacer es abrir el archivo, si llega a haber un problema te va a aparecer “error abriendo archivo” y si lo abre sin ningún error con ayuda de los comandos leerCadena, printf y scanf te va a empezar a pedir los datos que necesita de los estudiantes a ingresar como por ejemplo su legajo, el promedio ,su nombre y apellido, luego con los condicionales if y else nos va a crear un registro completo en el archivo binario y si esta todo bien le va a devolver 1 al programa y ahí empieza el if que dice que si sale todo bien nos va a devolver “estudiante registrado con éxito”, pero si llega a fallar el else nos va a devolver “error al escribir en el archivo”, por ultimo el fclose va a cerrar el archivo para que se guarde todo correctamente.

Listar estudiantes:



```
72
73 void listarEstudiantes() {
74     FILE *f = fopen(ARCHIVO_DATOS, "rb");
75     if (!f) { printf("Error abriendo archivo.\n"); return; }
76
77     Estudiante e;
78     int contador = 0;
79     printf("\n--- LISTADO DE ESTUDIANTES ACTIVOS ---\n");
80     while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
81         if (e.activo == 1) {
82             printf("Legajo: %d | Nombre: %s | Promedio: %.2f\n", e.legajo, e.nombre, e.promedio);
83             contador++;
84         }
85     }
86     if (contador == 0) printf("(sin estudiantes activos)\n");
87     fclose(f);
88 }
89
90 void buscarPorLegajo() {
```

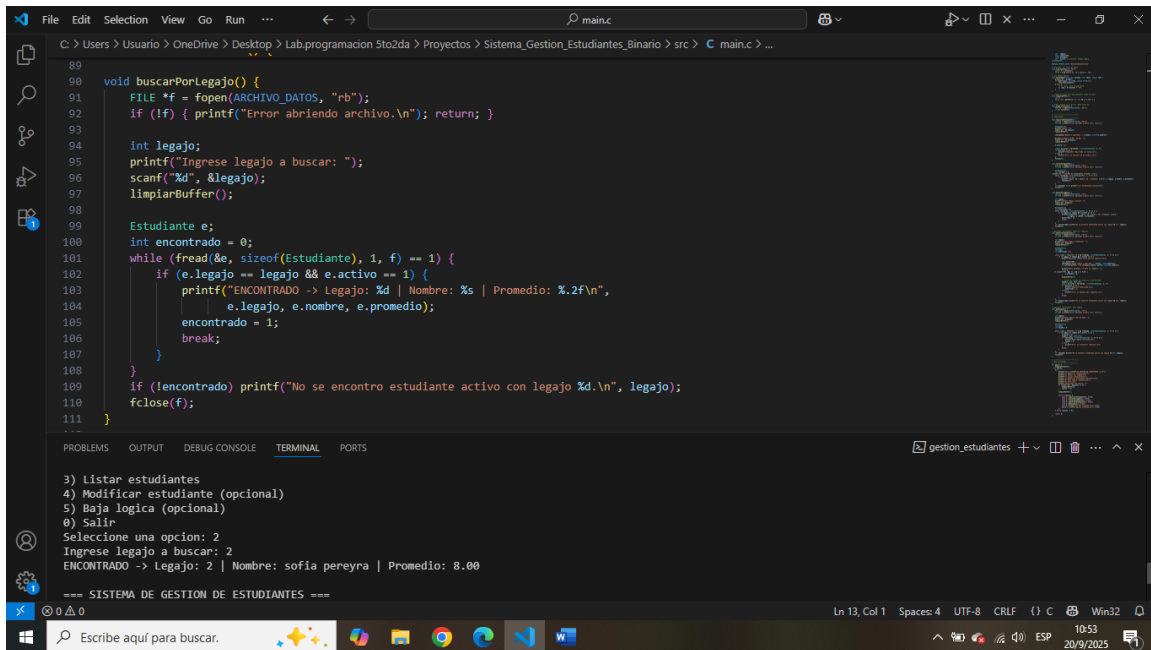
5) Baja logica (opcional)
0) Salir
Seleccione una opcion: 3

--- LISTADO DE ESTUDIANTES ACTIVOS ---
Legajo: 2 | Nombre: sofia pereyra | Promedio: 8.00

=== SISTEMA DE GESTION DE ESTUDIANTES ===
1) Registrar estudiante
2) Buscar por legajo
3) Listar estudiantes
4) Modificar estudiante (opcional)
5) Baja logica (opcional)
0) Salir
Seleccione una opcion: █

Esta es la opción 3 de el menú a lo largo de la imagen lo q hace este pedazo del código es abrir el archivo, si hay algún problema con eso el if te va a imprimir “error abriendo archivo” y si no lo hay el programa va a seguir asegurándose de que haya estudiantes activos si llega a haber estudiantes activos te los va a mostrar junto a su legajo, nombre y apellido, su promedio y esto pasa entre los comandos while y el primer if después del while el segundo if lo que hace es que si no llega a haber estudiantes activos va a leer 0 y te va a devolver “sin estudiantes activos”.

Buscar legajo:



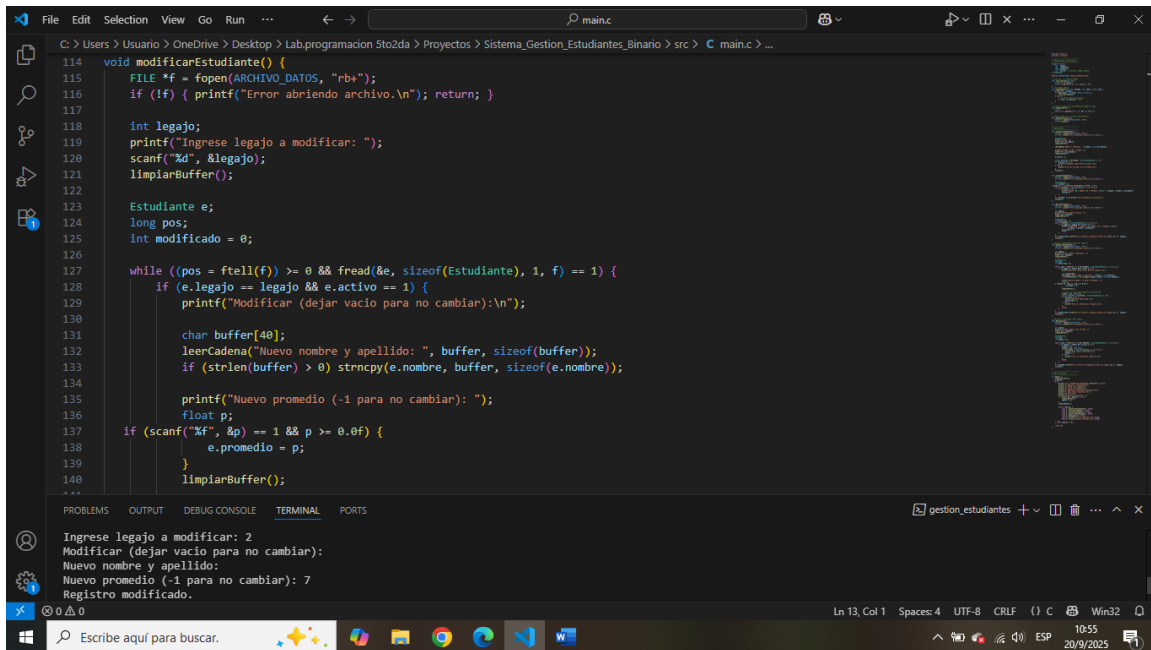
```
89
90 void buscarPorLegajo() {
91     FILE *f = fopen(ARCHIVO_DATOS, "rb");
92     if (!f) { printf("Error abriendo archivo.\n"); return; }
93
94     int legajo;
95     printf("Ingrese legajo a buscar: ");
96     scanf("%d", &legajo);
97     limpiarBuffer();
98
99     Estudiante e;
100    int encontrado = 0;
101    while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
102        if (e.legajo == legajo && e.activo == 1) {
103            printf("ENCONTRADO -> Legajo: %d | Nombre: %s | Promedio: %.2f\n",
104                e.legajo, e.nombre, e.promedio);
105            encontrado = 1;
106            break;
107        }
108    }
109    if (!encontrado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
110    fclose(f);
111 }
```

3) Listar estudiantes
4) Modificar estudiante (opcional)
5) Baja logica (opcional)
0) Salir
Seleccione una opcion: 2
Ingrese legajo a buscar: 2
ENCONTRADO -> Legajo: 2 | Nombre: sofia pereyra | Promedio: 8.00

=== SISTEMA DE GESTION DE ESTUDIANTES ===

En esta imagen esta la parte del código que pertenece a la opción 2 lo primero que hace es lo mismo que el anterior va a abrir el archivo, si hay algún problema con eso el if te va a imprimir “error abriendo archivo” y si no lo hay el programa va a seguir, te va a pedir que se ingrese un legajo el cual buscar con ayuda del printf y el scanf, luego con el while y el if va a buscar en el archivo un legajo que coincida con el que buscas además de que se va a asegurar de que este activo y si todo va bien te va a mostrar el nombre y apellido, el promedio del legajo que buscabas , en el caso de que no exista o no este activo el legajo de esa persona está el otro if que se va a encargar de mostrar en pantalla un mensaje que diga “no se encontró estudiante activo con legajo”.

Modificar estudiante:



```
114 void modificarEstudiante() {
115     FILE *f = fopen(ARCHIVO_DATOS, "rb+");
116     if (!f) { printf("Error abriendo archivo.\n"); return; }
117
118     int legajo;
119     printf("Ingrese legajo a modificar: ");
120     scanf("%d", &legajo);
121     limpiarBuffer();
122
123     Estudiante e;
124     long pos;
125     int modificado = 0;
126
127     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
128         if (e.legajo == legajo && e.activo == 1) {
129             printf("Modificar (dejar vacio para no cambiar):\n");
130
131             char buffer[40];
132             leerCadena("Nuevo nombre y apellido: ", buffer, sizeof(buffer));
133             if (strlen(buffer) > 0) strncpy(e.nombre, buffer, sizeof(e.nombre));
134
135             printf("Nuevo promedio (-1 para no cambiar): ");
136             float p;
137             if (scanf("%f", &p) == 1 && p >= 0.0f) {
138                 e.promedio = p;
139             }
140             limpiarBuffer();
141         }
142     }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

gestion_estudiantes + - [] ... ^ x

Ingrese legajo a modificar: 2
Modificar (dejar vacio para no cambiar):
Nuevo nombre y apellido:
Nuevo promedio (-1 para no cambiar): 7
Registro modificado.

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32 10:55 20/9/2025

En esta imagen el código pertenece a la opción 4 la cual es modificar estudiante básicamente lo primero que hace como en todas las opciones es abrir el archivo, si hay algún problema con eso el if te va a imprimir “error abriendo archivo” y si no lo hay el programa va a seguir, luego con ayuda del printf y scanf te va a pedir que ingreses un legajo del estudiante a modificar, luego con el while y el if va a fijarse que coincidan los legajos(el que pusiste y el que existe) y si coincide te va a pedir un nuevo nombre y apellido, un nuevo promedio


```
114 void modificarEstudiante() {
115     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
116         if (e.legajo == legajo && e.activo == 1) {
117             if (strlen(buffer) > 0) strncpy(e.nombre, buffer, sizeof(e.nombre));
118
119             printf("Nuevo promedio (-1 para no cambiar): ");
120             float p;
121             if (scanf("%f", &p) == 1 && p >= 0.0f) {
122                 e.promedio = p;
123             }
124             limpiarBuffer();
125
126             // volver al inicio del registro y sobrescribir
127             fseek(f, pos, SEEK_SET);
128             size_t escritos = fwrite(&e, sizeof(Estudiante), 1, f);
129             if (escritos == 1) {
130                 printf("Registro modificado.\n");
131                 modificado = 1;
132             } else {
133                 printf("Error al sobrescribir registro.\n");
134             }
135             break;
136         }
137     }
138     if (!modificado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
139     fclose(f);
140 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

gestion_estudiantes

Ingrese legajo a modificar: 2
Modificar (dejar vacio para no cambiar):
Nuevo nombre y apellido:
Nuevo promedio (-1 para no cambiar): 7
Registro modificado.

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32 10:56 20/9/2025

Esta imagen también pertenece a la opción 4 lo que hace desde el 143 al 151 es sobrescribir lo pedido en la anterior imagen en el estudiante y si encuentra ese archivo modificado el if va a mostrar un mensaje en pantalla que te avise que el registro se modifico y si falla el else te va a decir que hubo un error a la hora sobrescribir el registro todo esto en la misma ubicación en la que estaba ese archivo binario y el ultimo if es por si ese legajo que ingresaste no existe te va a imprimir un mensaje que diga “No se encontró estudiante activo con legajo 2” y luego el fclose se asegura de cerrar el archivo.

Baja lógica:

```
159 void bajaLogica() {
160     FILE *f = fopen(ARCHIVO_DATOS, "rb+");
161     if (!f) { printf("Error abriendo archivo.\n"); return; }
162
163     int legajo;
164     printf("Ingrese legajo a dar de baja: ");
165     scanf("%d", &legajo);
166     limpiarBuffer();
167
168     Estudiante e;
169     long pos;
170     int bajado = 0;
171
172     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
173         if (e.legajo == legajo && e.activo == 1) {
174             e.activo = 0;
175             fseek(f, pos, SEEK_SET);
176             if (fwrite(&e, sizeof(Estudiante), 1, f) == 1) {
177                 printf("Baja logica realizada.\n");
178                 bajado = 1;
179             } else {
180                 printf("Error al actualizar registro.\n");
181             }
182             break;
183         }
184     }
185     if (!bajado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
186     fclose(f);
187 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

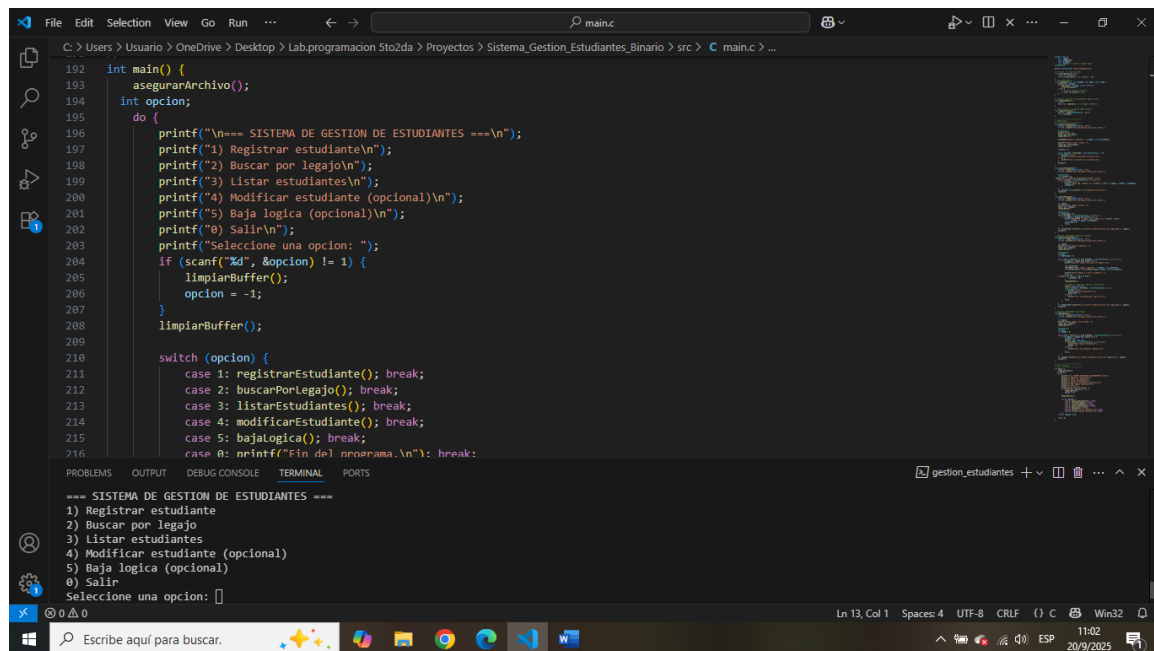
gestion_estudiantes

Seleccione una opcion: 5
Ingrese legajo a dar de baja: 2
Baja logica realizada.

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32 11:01 20/9/2025

En esta imagen está el código de la opción 5 como en las otras cuatro opciones lo primero que hace es abrir el archivo, si hay algún problema con eso el if te va a imprimir “error abriendo archivo” y si no lo hay el programa va a seguir luego con ayuda del printf y el scanf se pide ingresar un legajo el cual dar de baja luego con el while y el if se va a asegurar de que el legajo ingresado coincida con un estudiante activo (1) o lo contrario (0) y luego si coincide el if te va a imprimir un mensaje que avise que la baja lógica se realizó y sino el else te va a imprimir un mensaje que diga que hubo un error al actualizar el registro y el ultimo if sirve por si no hay ningún legajo que coincida con el que se ingresó y te va a regresar un mensaje que diga “no se encontró estudiante activo con legajo” y luego se cierra el archivo con el fclose como en las otras opciones.

Menú:

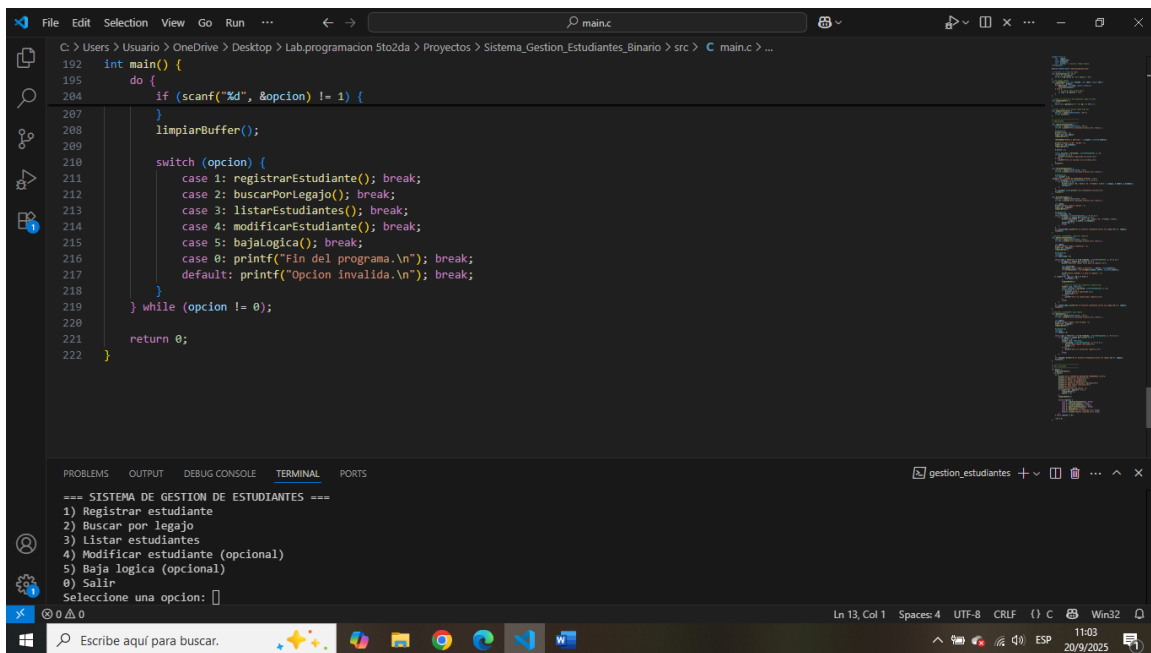


```
192 int main() {
193     asegurarArchivo();
194     int opcion;
195     do {
196         printf("\n=== SISTEMA DE GESTION DE ESTUDIANTES ===\n");
197         printf("1) Registrar estudiante\n");
198         printf("2) Buscar por legajo\n");
199         printf("3) Listar estudiantes\n");
200         printf("4) Modificar estudiante (opcional)\n");
201         printf("5) Baja logica (opcional)\n");
202         printf("0) Salir\n");
203         printf("Seleccione una opcion: ");
204         if (scanf("%d", &opcion) != 1) {
205             limpiarBuffer();
206             opcion = -1;
207         }
208         limpiarBuffer();
209
210         switch (opcion) {
211             case 1: registrarEstudiante(); break;
212             case 2: buscarPorLegajo(); break;
213             case 3: listarEstudiantes(); break;
214             case 4: modificarEstudiante(); break;
215             case 5: bajaLogica(); break;
216             case 0: printf("Fin del programa.\n"); break;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
=== SISTEMA DE GESTION DE ESTUDIANTES ===
1) Registrar estudiante
2) Buscar por legajo
3) Listar estudiantes
4) Modificar estudiante (opcional)
5) Baja logica (opcional)
0) Salir
Seleccione una opcion: 
```

En esta imagen junto a la de abajo lo que tenemos es el menú de las opciones que desde el 196 al 203 lo que hace es con el printf mostrar las distintas opciones disponibles, pero aclaro solo te las muestra luego con el if lo que hace es básicamente asegurarse de que se ingrese un numero valido (ej. 1, 2, 3, 0, etc.) y se ingresa algo que no es un numero (ej: carácter, letra) se limpia el buffer y con el “opción=-1” lo fuerza a ser un -1 para que luego en el switch lo tome como opción invalida.



```
192 int main() {
193     do {
204         if (scanf("%d", &opcion) != 1) {
205             limpiarBuffer();
206         }
207         switch (opcion) {
208             case 1: registrarEstudiante(); break;
209             case 2: buscarPorLegajo(); break;
210             case 3: listarEstudiantes(); break;
211             case 4: modificarEstudiante(); break;
212             case 5: bajaLogica(); break;
213             case 0: printf("Fin del programa.\n"); break;
214             default: printf("Opcion invalida.\n"); break;
215         } while (opcion != 0);
216     } while (opcion != 0);
217     return 0;
218 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

=== SISTEMA DE GESTION DE ESTUDIANTES ===
1) Registrar estudiante
2) Buscar por legajo
3) Listar estudiantes
4) Modificar estudiante (opcional)
5) Baja logica (opcional)
0) Salir
Seleccione una opcion: 0

En esta imagen seguimos en el código del menú y básicamente cada case lo que hace es compararte el numero ingresado por el usuario con cada opción (ej: 1, 2, etc.) y cuando la encuentra llama a esa función y luego con el break; hace que el programa no siga ejecutando los otros case y salga del switch.

El case 0 lo que hace es que si coincide con esta opción se va a imprimir un mensaje que diga “fin del programa”, y el default lo que hace es que si la opción no coincide con ninguna de las que estaban antes ejecuta este mensaje que va a decir que la opción es invalida.

Y el ultimo while lo que hace seria repetir todo el bloque de menú a menos que la opción sea 0.

GitHub:

El siguiente enlace te va a llevar al repositorio de github de Tiziana sandoval

https://github.com/TizianaW/Sistema_Gestion_Estudiantes.git