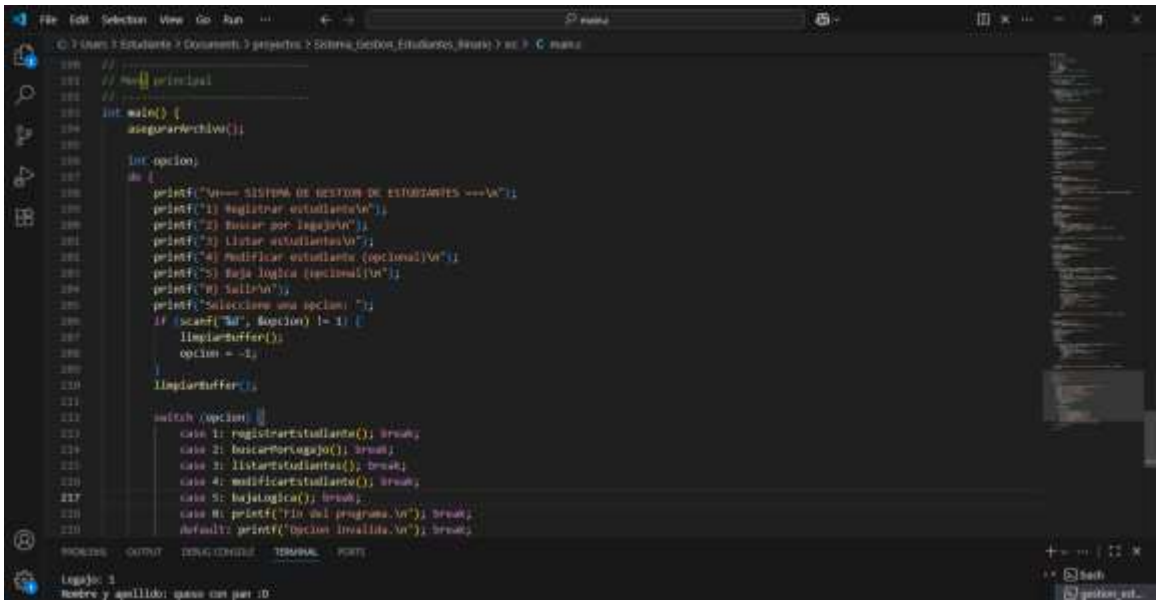
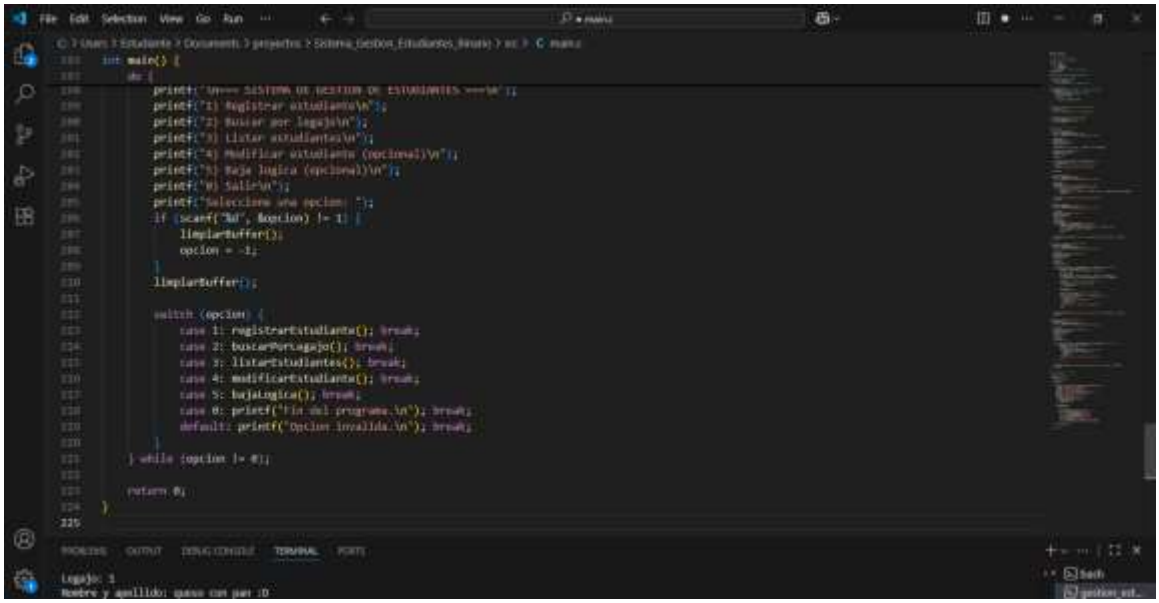
Sistema de Gestión de Estudiante

Nombre Completo del Estudiante: Kehit Nathaniel Fernandez Mori
Nombre de la Institución: Tecnica N°2
Curso: Laboratorio de Programación 5° 2° B
Profesor: Mansilla Muñoz York E.
Fecha: 20/09/2025

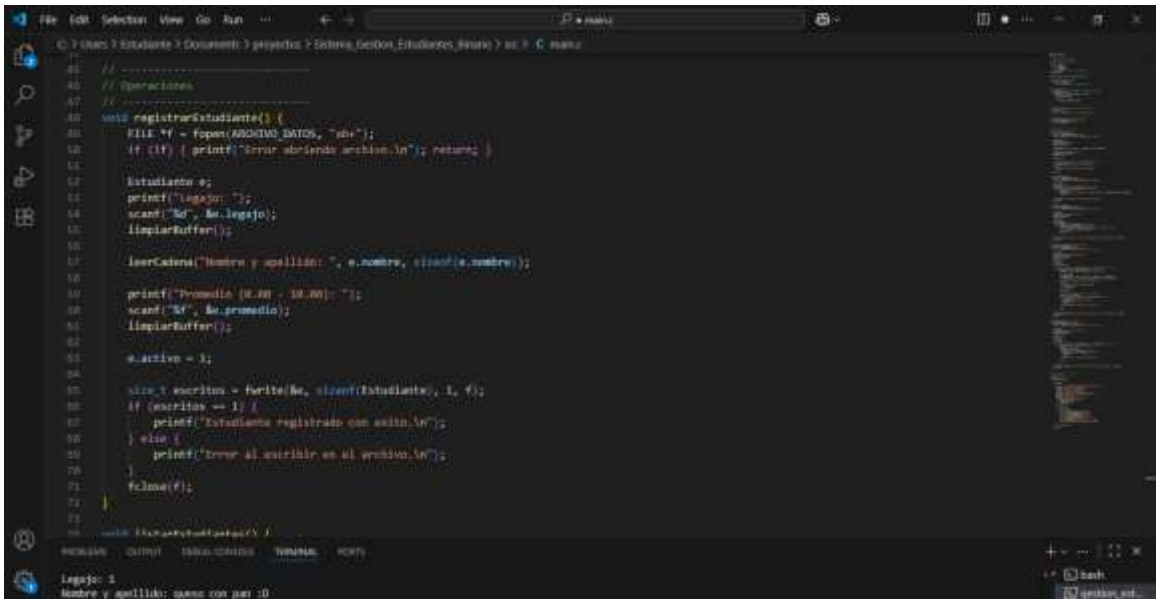
Menu Principal:

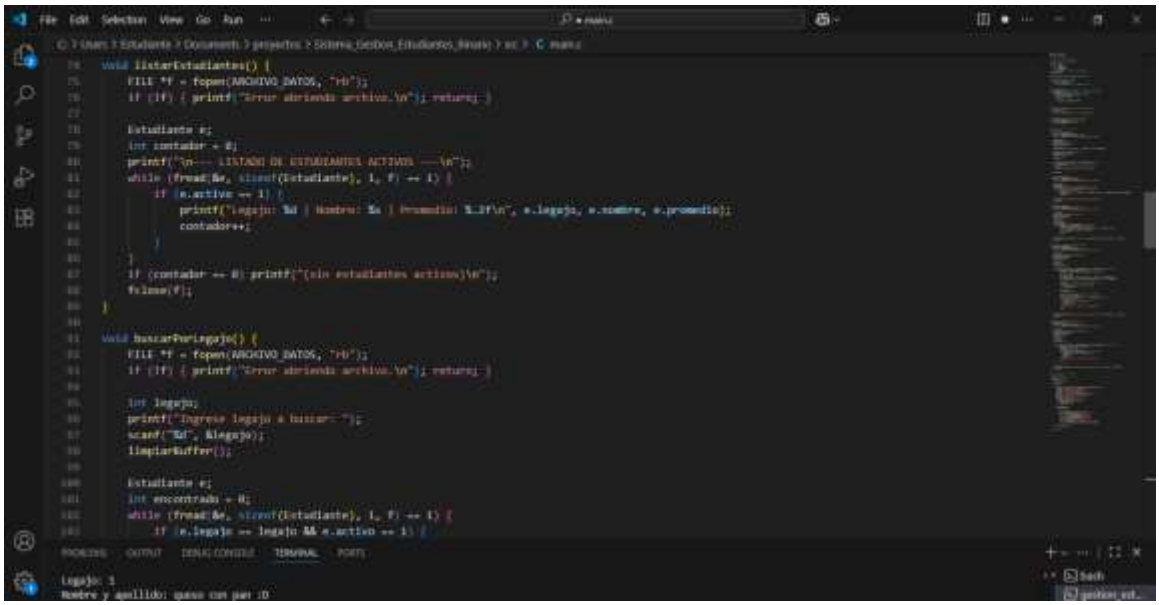


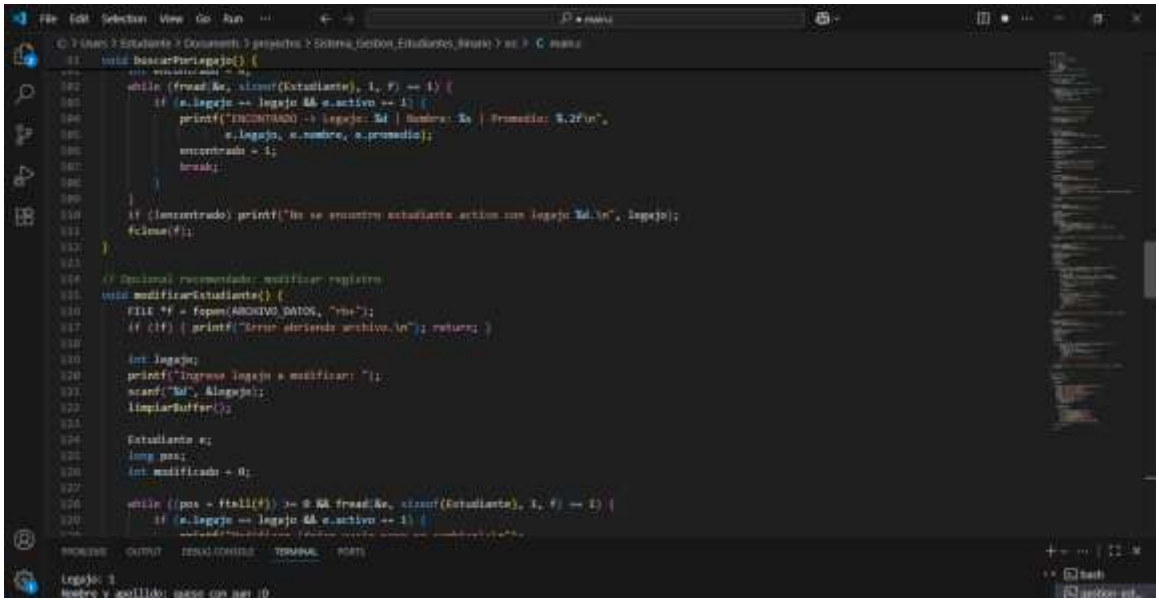


El programa del menu principal funciona de la siguiente manera: primero, se asegura de que entiendas las opciones que tienes disponibles, mostrandote un menu numerado. Luego, espera pacientemente a que ingreses el numero de la opcion que deseas. Para esto, utiliza la funcion scanf es importante que ingreses un numero, ya que el programa verifica si lo que escribiste es valido. Si te equivocas y escribes algo que no es un numero, el programa lo detecta y te dice que la opcion no es valida. Una vez que el programa tiene tu opcion (y se asegura de que sea valida), utiliza una estructura llamada switch para decidir que accion realizar. Cada numero del menu esta asociado a una accion especifica. Por ejemplo, si eliges la opcion 1, el programa ejecutara la funcion registrarEstudiante ( ).

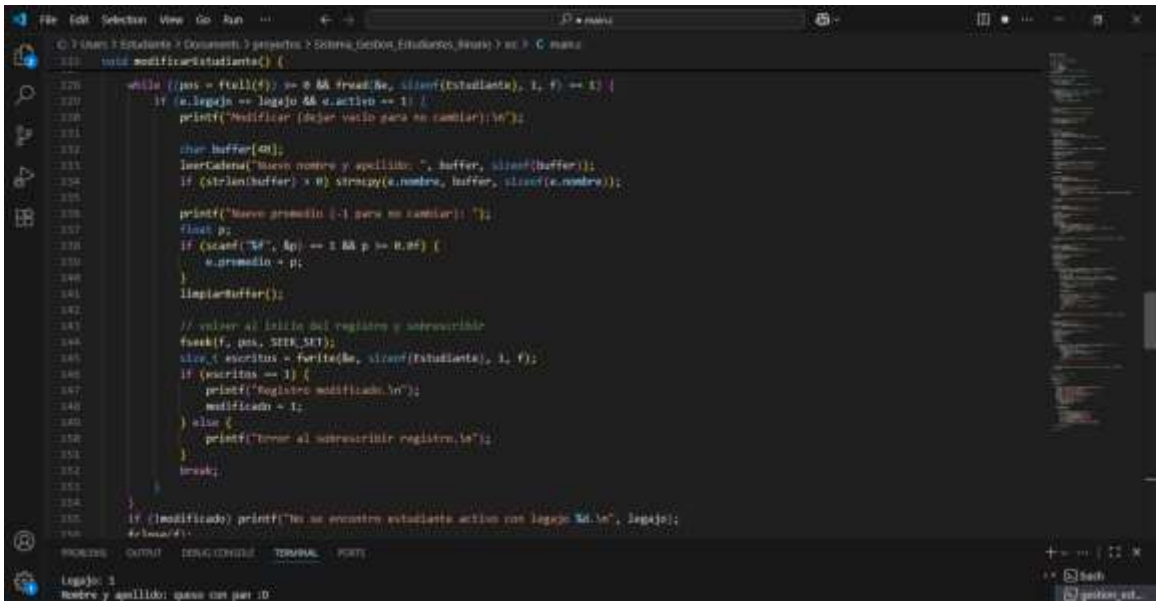
Operaciones:



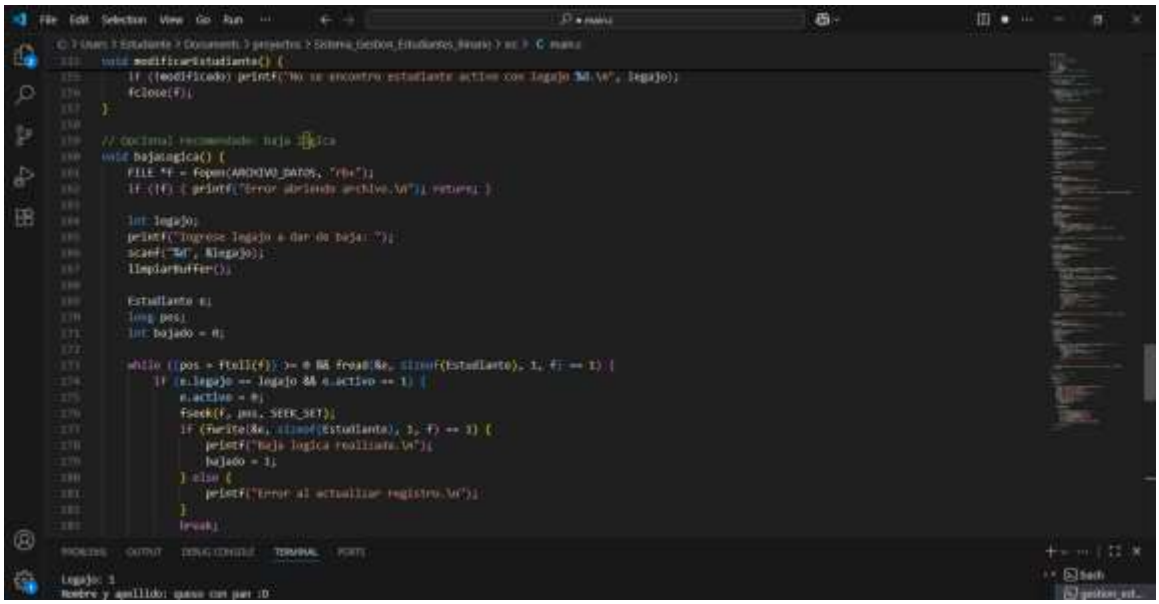


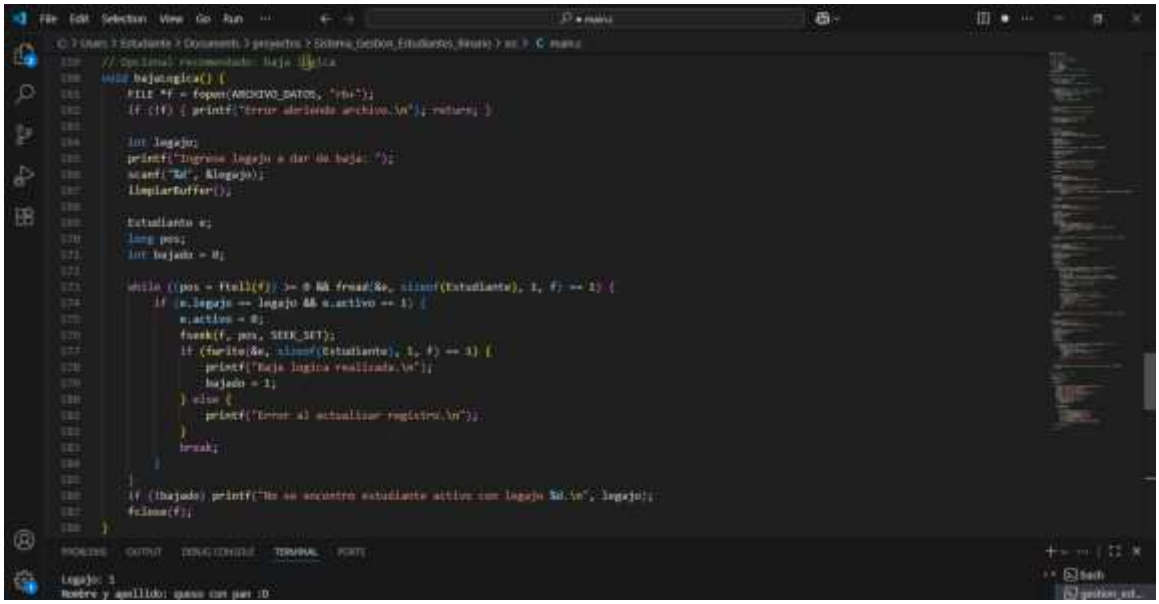


```
void BuscarPorLegajo() {
    while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
        if (e.legajo == legajo && e.activo == 1) {
            printf("ENCONTRADO -> Legajo: %d | Nombre: %s | Promedio: %.2f\n",
                   e.legajo, e.nombre, e.promedio);
            encontrado = 1;
            break;
        }
    }
    if (!encontrado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
    fclose(f);
}

// Opcional recomendado: modificar registro
void modificarEstudiante() {
    FILE *f = fopen(ARCHIVO_DATOS, "rb+");
    if (!f) { printf("Error abriendo archivo.\n"); return; }

    int legajo;
    printf("Ingrese legajo a modificar: ");
    scanf("%d", &legajo);
    limpiarBuffer();

    Estudiante e;
    long pos;
    int modificado = 0;

    while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
        if (e.legajo == legajo && e.activo == 1) {
```



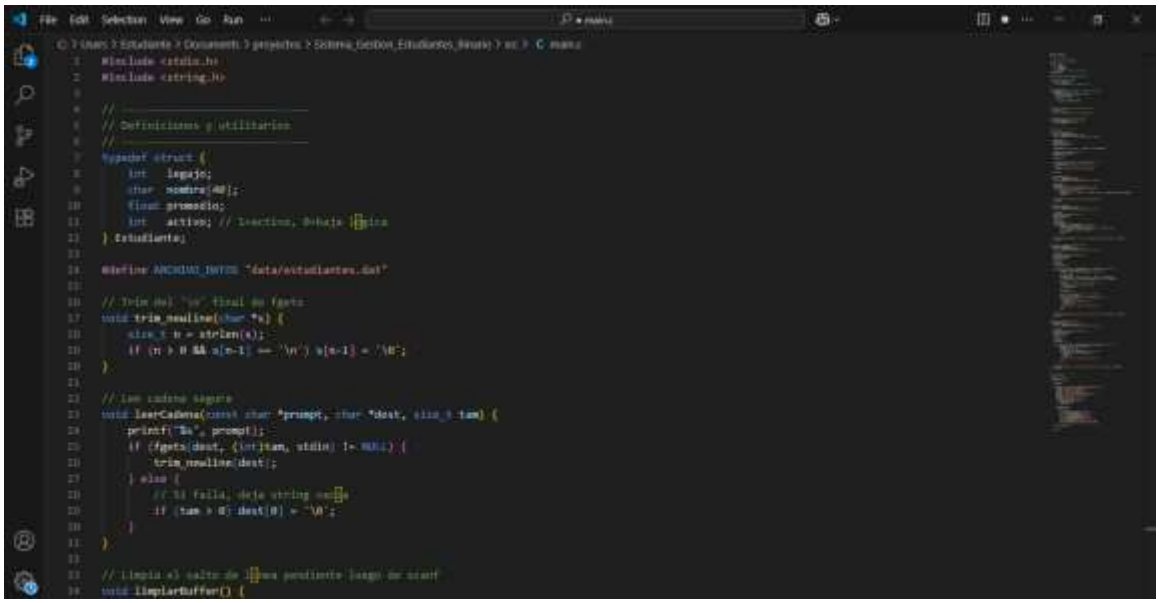
```
void modificarEstudiante() {
    while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
        if (e.legajo == legajo && e.activo == 1) {
            printf("Modificar (dejar vacio para no cambiar):\n");

            char buffer[40];
            leerCadena("Nuevo nombre y apellido: ", buffer, sizeof(buffer));
            if (strlen(buffer) > 0) strcpy(e.nombre, buffer, sizeof(e.nombre));

            printf("Nuevo promedio (-1 para no cambiar): ");
            float p;
            if (scanf("%f", &p) == 1 && p >= 0.0f) {
                e.promedio = p;
            }
            limpiarBuffer();

            // volver al inicio del registro y sobrescribir
            fseek(f, pos, SEEK_SET);
            size_t escritos = fwrite(&e, sizeof(Estudiante), 1, f);
            if (escritos == 1) {
                printf("Registro modificado.\n");
                modificado = 1;
            } else {
                printf("Error al sobrescribir registro.\n");
            }
            break;
        }
    }
    if (!modificado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
    fclose(f);
}
```

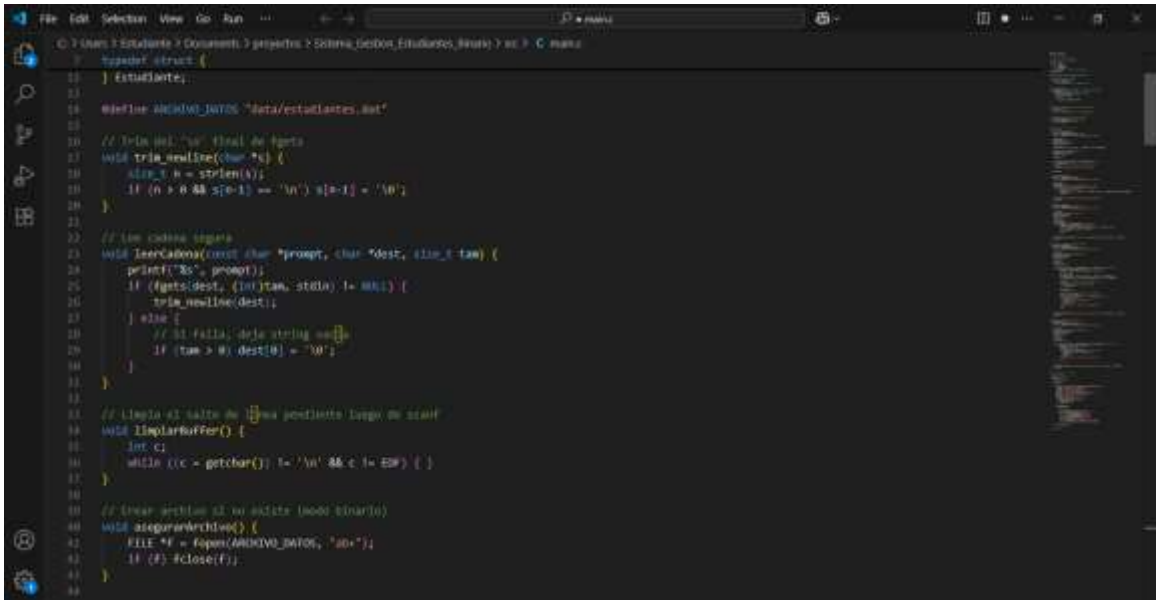




El Código implementa un sistema de gestion de estudiantes que utiliza un archivo binario para almacenar la informacion de cada estudiante. Cada estudinte tiene un legajo (ID), nombre, apellido, promedio y un indicador de si esta activo o no. las funciones principales permiten registrar nuevos estudiantes, listar los estudiantes activos, buscar un estudiante por su legajo, modificar la informacion de un estudiante y realizar una baja logica (marcar un estudiante como inactivo). Ademas emplea funciones clave de manejo de archivo: fopen para abrir el archivo en el modo adecuado, fread para leer registros y cargarlos en memoria fwrite para escribir nuevos registros o modificar existentes, fseek para mover el puntero de lectura/escritura a posiciones especificas dentro del archivo y ftell para obtener la posicion actual del puntero. El flujo de operaciones incluye abrir el archivo, leer o modificar registros utilizando fseek y las funciones de lectura/escritura, y finalmente cerrar el archivo con fclose, asegurando una gestion eficiente de la informacion de estudiantes.

Definiciones y Utilitarios:





Defini una estructura llamada Estudiante que es como un formulario con los datos básicos: legajo, nombre, promedio y si está activo. Luego, con funciones Como leerCadena para leer nombres de forma segura y limpiarBuffer para evitar errores al leer del teclado. archivo donde guardarán los datos exista, con la función crearArchivo . En conclusión, es un código bien pensado para sentar las bases de un programa que gestione información de estudiantes de forma organizada y sin errores comunes.

Este es el link de github de kehit fernandez mori:

https://github.com/kehit19/Sistema_Gestion_Estudiante.git