

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:

```
198 // menu principal
199 //
200 int main() {
201     asegurarArchivo();
202
203     int opcion;
204     do {
205         printf("\n== SISTEMA DE GESTION DE ESTUDIANTES ==\n");
206         printf("1) Registrar estudiante\n");
207         printf("2) Buscar por legajo\n");
208         printf("3) Listar estudiantes\n");
209         printf("4) Modificar estudiante (opcional)\n");
210         printf("5) Baja logica (opcional)\n");
211         printf("6) Salir\n");
212         printf("Seleccione una opcion: ");
213         if (scanf("%d", &opcion) != 1) {
214             limpiarBuffer();
215             opcion = -1;
216         }
217         limpiarBuffer();
218
219         switch (opcion) {
220             case 1: registrarEstudiante(); break;
221             case 2: buscarPorLegajo(); break;
222             case 3: listarEstudiantes(); break;
223             case 4: modificarEstudiante(); break;
224             case 5: bajaLogica(); break;
225             case 6: printf("Fin del programa.\n"); break;
226             default: printf("Opcion invalida.\n"); break;
227         }
228     } while (opcion != 6);
229
230     return 0;
231 }
```

```
198 // menu principal
199 //
200 int main() {
201     do {
202         printf("\n== SISTEMA DE GESTION DE ESTUDIANTES ==\n");
203         printf("1) Registrar estudiante\n");
204         printf("2) Buscar por legajo\n");
205         printf("3) Listar estudiantes\n");
206         printf("4) Modificar estudiante (opcional)\n");
207         printf("5) Baja logica (opcional)\n");
208         printf("6) Salir\n");
209         printf("Seleccione una opcion: ");
210         if (scanf("%d", &opcion) != 1) {
211             limpiarBuffer();
212             opcion = -1;
213         }
214         limpiarBuffer();
215
216         switch (opcion) {
217             case 1: registrarEstudiante(); break;
218             case 2: buscarPorLegajo(); break;
219             case 3: listarEstudiantes(); break;
220             case 4: modificarEstudiante(); break;
221             case 5: bajaLogica(); break;
222             case 6: printf("Fin del programa.\n"); break;
223             default: printf("Opcion invalida.\n"); break;
224         }
225     } while (opcion != 6);
226
227     return 0;
228 }
```

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:

```

1 // .....
2 // Operaciones
3 // .....
4 void registrarEstudiante() {
5     FILE *f = fopen(ABOQUIVO_DATOS, "a+");
6     if (!f) { printf("Error abriendo archivo.\n"); return; }
7
8     Estudiante e;
9     printf("legajo: ");
10    scanf("%d", &e.legajo);
11    limpiarBuffer();
12
13    printf("Nombre y apellido: ", e.nombre, sizeof(e.nombre));
14
15    printf("Promedio (0.00 - 10.00): ");
16    scanf("%f", &e.promedio);
17    limpiarBuffer();
18
19    e.activo = 1;
20
21    size_t escritas = fwrite(&e, sizeof(Estudiante), 1, f);
22    if (escritas == 1) {
23        printf("Estudiante registrado con exito.\n");
24    } else {
25        printf("Error al escribir en el archivo.\n");
26    }
27    fclose(f);
28 }
29
30 .....
31
32 legajo: 1
33 Nombre y apellido: paco con nro :0

```

```

14 void listarEstudiantes() {
15     FILE *f = fopen(ARCHIVO_DATOS, "r");
16     if (!f) { printf("Error abriendo archivo.\n"); return; }
17
18     Estudiante e;
19     int contador = 0;
20     printf("\n--- LISTADO DE ESTUDIANTES ACTIVOS ---\n");
21     while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
22         if (e.activo == 1) {
23             printf("legajo: %d | nombre: %s | promedio: %.2f\n", e.legajo, e.nombre, e.promedio);
24             contador++;
25         }
26     }
27     if (contador == 0) printf("sin estudiantes activos\n");
28     fflush(f);
29 }
30
31 void buscarPorLegajo() {
32     FILE *f = fopen(ARCHIVO_DATOS, "r");
33     if (!f) { printf("Error abriendo archivo.\n"); return; }
34
35     int legajo;
36     printf("Ingrese legajo a buscar: ");
37     scanf("%d", &legajo);
38     limpiarBuffer();
39
40     Estudiante e;
41     int encontrado = 0;
42     while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
43         if (e.legajo == legajo && e.activo == 1) {

```

```

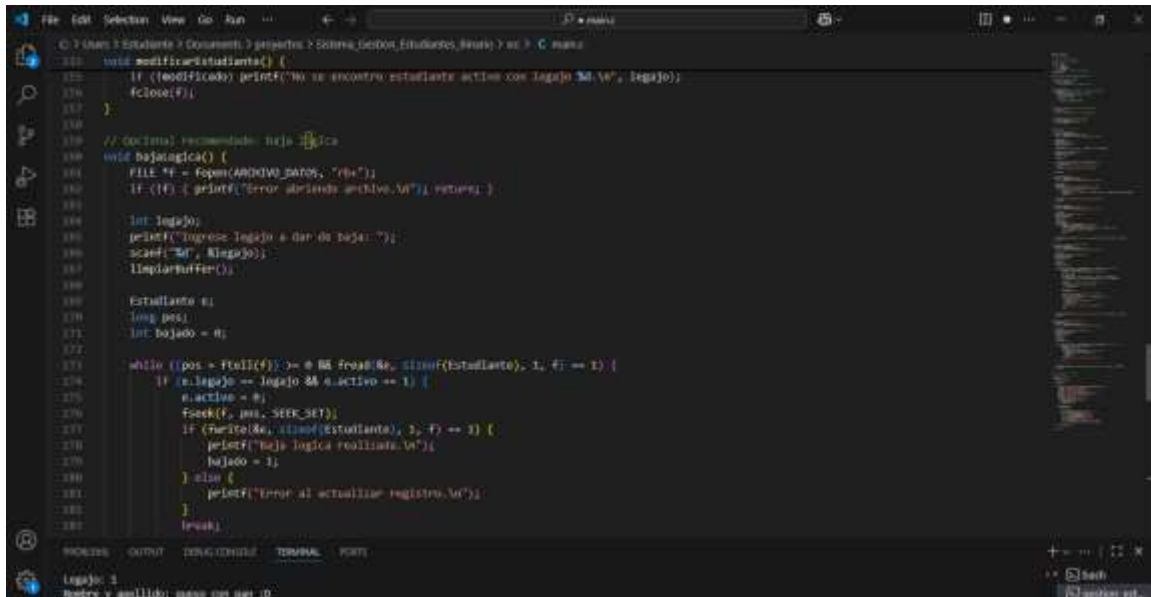
101 void buscarPorLapso() {
102     int encontrado = 0;
103     while (fread(&e, sizeof(Estudiante), 1, f) == 1) {
104         if (e.lapso == lapso || e.activo == 1) {
105             printf("ENCUENTRO -> Lapso: %d | Nombre: %s | Promedio: %.2f\n",
106                 e.lapso, e.nombre, e.promedio);
107             encontrado = 1;
108             break;
109         }
110     }
111     if (!encontrado) printf("No se encontro estudiante activo con lapso %d\n", lapso);
112     fclose(f);
113 }
114
115 // Opcional: recomendar: modificar registro
116 void modificarEstudiante() {
117     FILE *f = fopen(ARCHIVO_DATOS, "r+");
118     if (!f) { printf("Error abriendo archivo.\n"); return; }
119
120     int lapso;
121     printf("Ingrese lapso a modificar: ");
122     scanf("%d", &lapso);
123     limpiarBuffer();
124
125     Estudiante e;
126     long pos;
127     int modificado = 0;
128
129     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
130         if (e.lapso == lapso && e.activo == 1) {

```

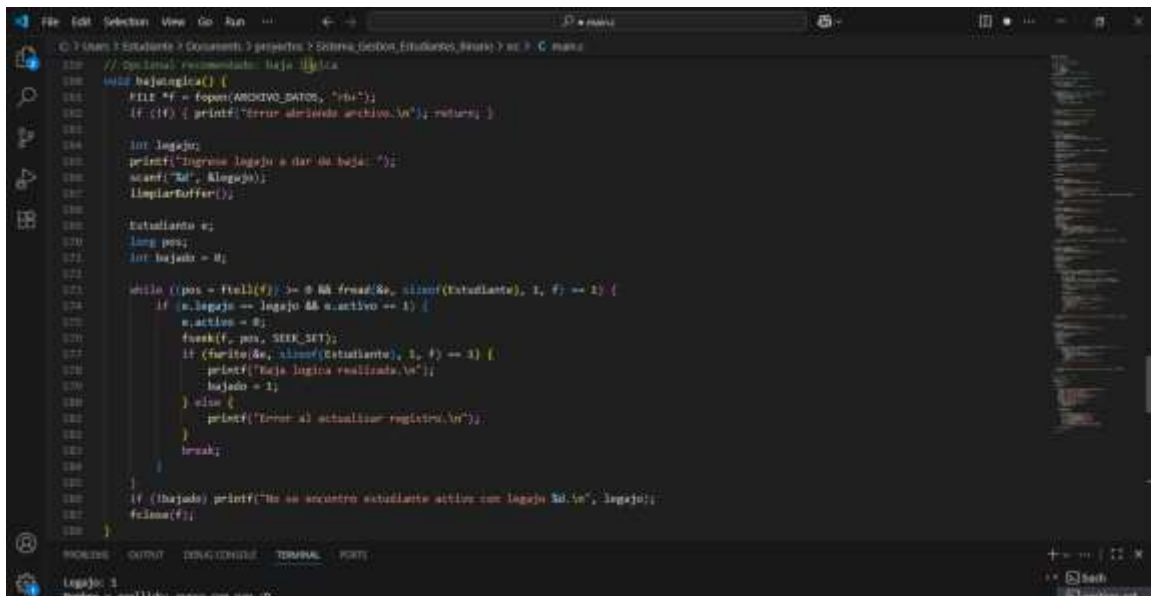
```

131         void modificarEstudiante() {
132             while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
133                 if (e.lapso == lapso && e.activo == 1) {
134                     printf("Modificar (dejar vacío para no cambiar):\n");
135
136                     char buffer[60];
137                     InsertarData("Nuevo nombre y apellido: ", buffer, sizeof(buffer));
138                     if (strlen(buffer) > 0) strcpy(e.nombre, buffer, sizeof(e.nombre));
139
140                     printf("Nuevo promedio (-1 para no cambiar): ");
141                     float p;
142                     if (scanf("%f", &p) == 1 && p >= 0.0f) {
143                         e.promedio = p;
144                     }
145                     limpiarBuffer();
146
147                     // volver al inicio del registro y sobrescribir
148                     fseek(f, pos, SEEK_SET);
149                     size_t escritos = fwrite(&e, sizeof(Estudiante), 1, f);
150                     if (escritos == 1) {
151                         printf("Registro modificado.\n");
152                         modificado = 1;
153                     } else {
154                         printf("Error al sobrescribir registro.\n");
155                     }
156                     break;
157                 }
158             }
159
160             if (!modificado) printf("No se encontro estudiante activo con lapso %d\n", lapso);
161             fclose(f);

```



```
121 void modificarEstudiante() {
122     if (!modificado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
123     fclose(f);
124 }
125
126 // Opcional: recomendacion: baja logica
127 void bajaLogica() {
128     FILE *f = fopen(ARCHIVO_DATOS, "rb+");
129     if (!f) { printf("Error abriendo archivo.\n"); return; }
130
131     int legajo;
132     printf("Ingrese legajo a dar de baja: ");
133     scanf("%d", &legajo);
134     fflush(stdin);
135
136     Estudiante e;
137     long pos;
138     int bajado = 0;
139
140     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
141         if (e.legajo == legajo && e.activo == 1) {
142             e.activo = 0;
143             fseek(f, pos, SEEK_SET);
144             if (fwrite(&e, sizeof(Estudiante), 1, f) == 1) {
145                 printf("Baja logica realizada.\n");
146                 bajado = 1;
147             } else {
148                 printf("Error al actualizar registro.\n");
149             }
150         }
151     }
152     break;
153 }
```



```
120 // Opcional: recomendacion: baja logica
121 void bajaLogica() {
122     FILE *f = fopen(ARCHIVO_DATOS, "rb+");
123     if (!f) { printf("Error abriendo archivo.\n"); return; }
124
125     int legajo;
126     printf("Ingrese legajo a dar de baja: ");
127     scanf("%d", &legajo);
128     fflush(stdin);
129
130     Estudiante e;
131     long pos;
132     int bajado = 0;
133
134     while ((pos = ftell(f)) >= 0 && fread(&e, sizeof(Estudiante), 1, f) == 1) {
135         if (e.legajo == legajo && e.activo == 1) {
136             e.activo = 0;
137             fseek(f, pos, SEEK_SET);
138             if (fwrite(&e, sizeof(Estudiante), 1, f) == 1) {
139                 printf("Baja logica realizada.\n");
140                 bajado = 1;
141             } else {
142                 printf("Error al actualizar registro.\n");
143             }
144         }
145     }
146     break;
147 }
148
149 if (!bajado) printf("No se encontro estudiante activo con legajo %d.\n", legajo);
150 fclose(f);
151 }
```

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:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 // Definiciones y utilitarios
5
6 typedef struct {
7     int legajo;
8     char nombre[40];
9     float promedio;
10    int activo; // 1=activo, 0=inactivo
11 } Estudiante;
12
13 #define ARCHIVO_DATOS "data/estudiantes.dat"
14
15 // Función para leer una cadena de forma segura
16 void leerCadena(char *dest, int tam) {
17     printf("Ingrese nombre: ");
18     if (fgets(dest, (int)tam, stdin) != NULL) {
19         trim_newline(dest);
20     } else {
21         // Si falla, deja string vacío
22         if (tam > 0) dest[0] = '\0';
23     }
24 }
25
26 // Limpia el buffer de entrada para evitar errores al leer del teclado
27 void limpiarBuffer() {
28     int c;
29     while ((c = getchar()) != '\n' && c != EOF) {
30     }
31 }
```

```
1 // Limpia el buffer de entrada para evitar errores al leer del teclado
2 void limpiarBuffer() {
3     int c;
4     while ((c = getchar()) != '\n' && c != EOF) {
5     }
6 }
7
8 // Función para crear el archivo de datos
9 void crearArchivo() {
10    FILE *f = fopen(ARCHIVO_DATOS, "a+");
11    if (!f) {
12        printf("Error al crear el archivo.\n");
13    }
14    fclose(f);
15 }
```

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