

Universidad de Guanajuato División de Ingenierías
Campus Irapuato Salamanca (DICIS)

Algoritmos y estructura de datos
Carlos Hugo García Capulín

Tarea No. 6
Reporte de Registro de Calificaciones.

Jair Chávez Islas
14/Octubre/2021

Problema

En esta tarea, tenemos el programa que hemos estado haciendo con el maestro en clase, sobre el registro de calificaciones, ya tiene sus funciones como ingresar registro, para capturar las calificaciones del alumno en cada materia, por ejemplo, guardar esos registros y después mostrarlo o ver cuántos registros se hicieron; Para esta tarea, programaré 3 funciones más, las cuales son promedio general, promedio por semestre y total de créditos cursados, estos datos los sacaré en base a los registros que ingrese el usuario.

Para sacar el promedio general, pues como en cualquier situación, se suman las calificaciones y se divide entre el número de materias, así que tenemos el siguiente ejemplo.

Alguien sacó las siguientes calificaciones en algunas udas.

10, 7, 5, 8, 9 y 8; entonces en este caso, sumamos las calificaciones

$$10 + 7 + 5 + 8 + 9 + 8 = 47$$

$47/6 = 7.8333$ y ese sería su promedio.

Para sacar el total de créditos, necesitamos recopilar los créditos de las materias que se están capturando y sumarlos, pero ojo que, si la materia no se ha aprobado, no se pueden sumar esos créditos.

Tenemos las primeras 6 materias, suponiendo que son las mismas materias y calificaciones que se sacaron en el promedio final, las cuales sus créditos son de 5, 8, 6, 3, 6, 6.

$5 + 8 + 6 + 3 + 6 + 6 = 34$, en teoría el resultado es 34, pero como vimos, en su tercer materia sacó 5, así que esos 6 créditos no deberían sumarse, dando como resultado 29 créditos aprobados.

Solución implementada

Diagrama de flujo de la primera funcion agregada

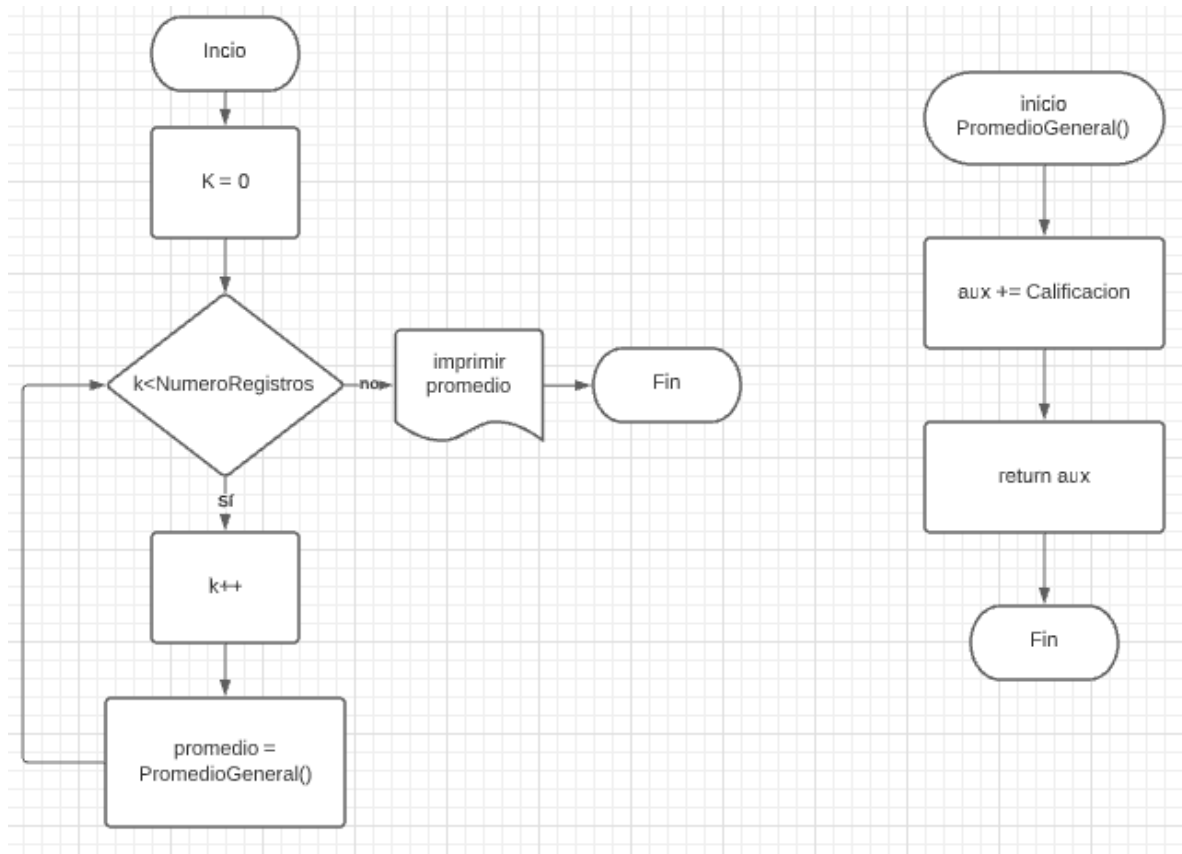
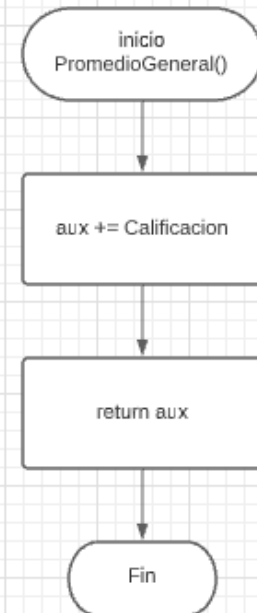
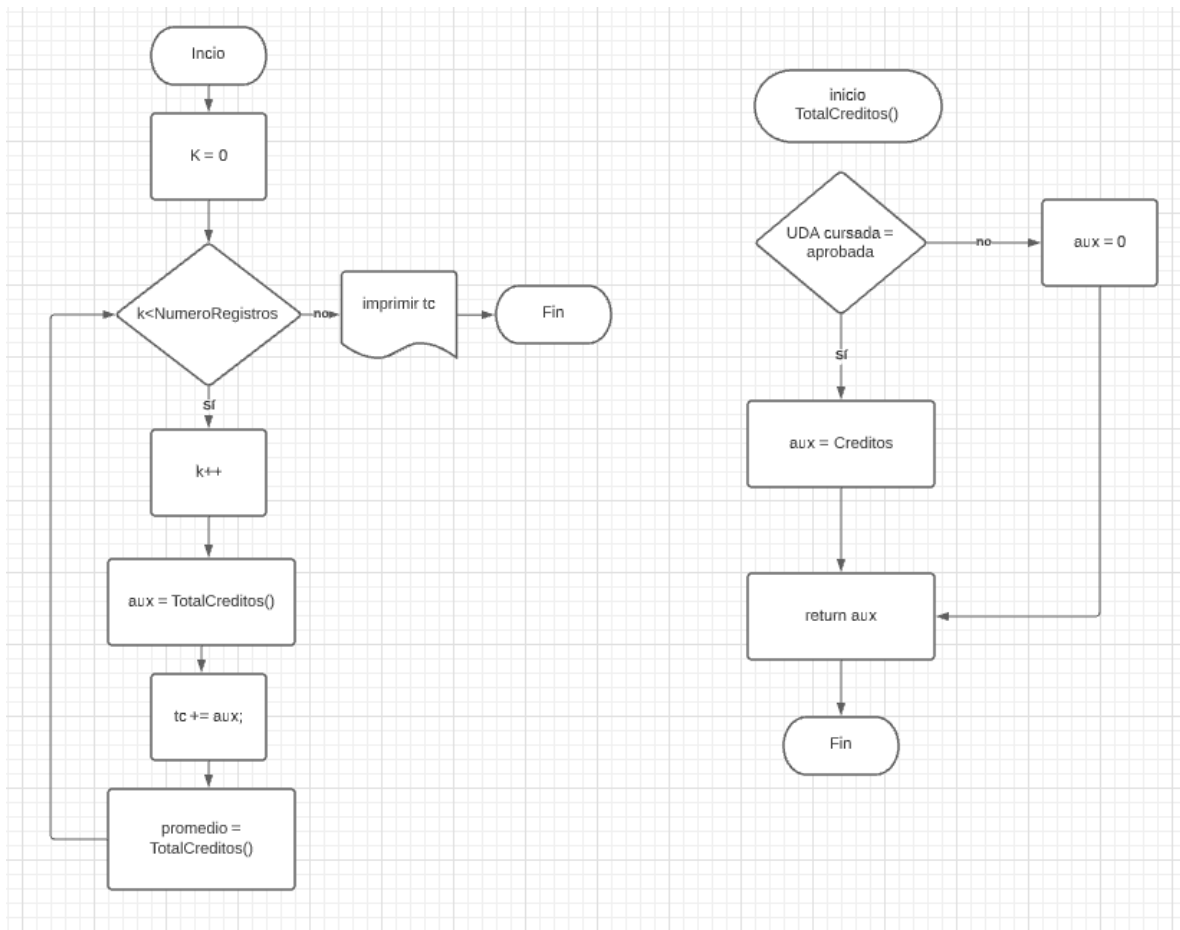


Diagrama de flujo de la segunda funcion agregada





Tomando las udas de donde las estuvimos tomando desde el principio, las usaremos nuevamente para comprobar el número de créditos

Código comentado de la primera función agregada

```

float PromedioGeneral(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])
{
    //declaramos una variable float llamada aux
    float aux;

    // aqui se van sumando las calificaciones que ya registró el usuario
    aux = aux + UDAcursada[NumReg-1].Status.Calificacion;

    //se retorna la variable aux
    return aux;
}

```

Para lograr sacar el promedio de las calificaciones, me fui guiando del código del maestro, especialmente de la función PrintRegistro(), ya que vi, que de ahí obtenía la calificación que habíamos ingresado, para luego imprimirla, así que hice más o menos lo mismo y con una variable aux, iba sumando esa calificación.

```

case 5:
//Se inicia un ciclo for que repasa los registros que se han hecho
for(k=0; k<NumeroRegistros; k++)
//aqui con la variable promedio, mandamos a llamar a la funcion y luego
//la imprimimos
promedio = (PromedioGeneral(UDAcursada,k+1,UDA))/NumeroRegistros;
printf("\n El promedio general es: %f\n",promedio);
break;

```

En esta parte se mandó a llamar la función, esta parte del código intenté hacerla varias veces dentro de la función PromedioGeneral() pero me imprimía valor muy muy muy grandes, así que, la forma en que em salió fue así, entonces el for hace una recopilación de los registros ingresados y con la variable promedio, mandamos a llamar la función del promedio general y lo dividimos entre el número de registros, ya que cada registro es una calificación, y eso da el promedio, y después imprimimos al variable promedio.

Código comentado de la segunda función agregada

```

//Aqui se inicia la funcion para sacar el total de creditos
int TotalCreditos(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])
{
    //decalramos una variable aux
    unsigned int aux=0;

    //abrimos un if, en el que si la materia está aprobada, aux vale la cantidad de creditos
    if(UDAcursada[NumReg-1].Status.Aprobada)
    {
        aux = ((int)(ListadoUDAS[NumReg-1].NumCre));
    }
    //en caso de que la materia no haya sido aprobada, los creditos no se suman
    else
    {
        aux = 0;
    }

    return aux;
}

```

En esta función me basé mucho en la función hecha por el maestro PrintUDA() y de PrintRegistro() ya que usando estas funciones se imprime el número de créditos de la materia correspondiente, entonces haciéndolo más o menos igual, saqué el número de créditos de las materias ingresadas, pero en un ciclo if, puse que, si la materia está aprobada, la variable aux, vale lo mismo que la cantidad de créditos, pero si la materia no está aprobada, la variable aux vale 0.

```

case 7:
//Se inciaa un ciclo for que repasa los registros que se han hecho
for(k=0; k<NumeroRegistros; k++)
{
    //Con la variable aux, llamo a la función
    aux = TotalCreditos(UDAcursada,k+1,UDA);
    //voy sumando los resultados y los guardo en la variable tc
    tc += aux;
}

//imprimo el total de creditos
printf("\nTotal de creditos: %u",tc);
break;

```

Se abre un ciclo for en el que se hace repaso de los registros hechos por el usuario, aquí hay una variable llamada tc (totalcreditos), y dentro de ese ciclo, la variable aux, llama a la función anterior y todo se va sumando en la variable tc, para luego imprimirla.

Lamentablemente para este reporte no pude hacer la función de promedio por semestre, por falta de tiempo e ingenio.

Código completo comentado

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  //Se declara una estructura llamada UNIDAD DE APRENDIZAJE
5  //con los miembros Id, Nombre, Area, Tipo, NumCre y HorasSem
6  typedef struct
7  {
8      unsigned char Id;
9      char Nombre[100];
10     char Area[50];
11     char Tipo[50];
12     unsigned int NumCre;
13     unsigned char HorasSem;
14 }UNIDADdeAPRENDIZAJE;
15
16 //Se declara una estructura llamada UDA_ESTATUS
17 //con los miembros Calificacion, oportunidad y aprobada
18 typedef struct
19 {
20     float Calificacion;
21     unsigned int Oportunidad;
22     unsigned char Aprobada;
23 }UDA_ESTATUS;
24
25 //Se declara una estructura llamada PERSONA
26 //con los miembros Nombres, Apaterno y Amaterno
27 typedef struct
28 {
29     char Nombres[40];
30     char Apaterno[20];
31     char Amaterno[20];
32 }PERSONA;
33
34 //Se declara una estructura llamada REGISTRO
35 //con los miembros Id, Id_UDA y tiene 2 miembros que son
36 //de tipo estructura als caules ya declaramos antes
37 typedef struct
38 {
39     unsigned char Id;
40     unsigned char Id_UDA;
41     UDA_ESTATUS Status;
42     PERSONA Profesor;
43 }REGISTRO;
44
45 //Aqui van las variables globales, aqui usamos la variable del nombre del archivo
46 //para hacerlo mas facil de abrir, ese archivo es de donde sacamos toda la
47 //informacion de las udas con la que hemos estado trabajndo
48 const char NombreDelArhivoDeUDAS[]="ListadoUDA.txt";
49 //como en el archivo hay 72 udas, tambien declaramos la variable global
50 //del total de udas para que sea mas facil de acceder.
```

```

50 //del total de udas para que sea mas facil de acceder.
51 const unsigned char Total_UDAS=72;
52
53 //Aqui van los prototipos d elas funciones que se utilizaron durante el programa
54 void PrintUDA(UNIDADdeAPRENDIZAJE UDA);
55 void LecturaArchivoUDAS(UNIDADdeAPRENDIZAJE UDA[]);
56 void CapturarRegistro(REGISTRO UDAcursada[], unsigned char NumReg);
57 void PrintRegistro(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[]);
58 void GuardarRegistros(REGISTRO UDAcursada[], unsigned int NumDeRegistros);
59 unsigned char LeerRegistros(REGISTRO UDAcursada[]);
60 float PromedioGeneral(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[]);
61 int TotalCreditos(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[]);
62 void listadou(UNIDADdeAPRENDIZAJE UDA[]);
63 void Printuda(UNIDADdeAPRENDIZAJE UDA);
64
65 int main()//se inicializa la funcion principal
66 {
67     //Se declara un Arreglo de estructuras de datos de tipo UNIDAD DE APRENDIZAJE llamado UDA
68     UNIDADdeAPRENDIZAJE UDA[Total_UDAS];
69     //Se declara un Arreglo de estructuras de datos de tipo REGISTRO llamado UDAcursada
70     REGISTRO UDAcursada[Total_UDAS];
71     //aqui hay variables varias que utilizamos en la funcion principal
72     unsigned char NumeroRegistros=0;
73     unsigned char salir=0;
74     unsigned int opcion,k;
75     float promedio;
76     int tc = 0;
77     int aux;
78
79     //Se manda a llamar la función que lee el archivo, se ahce desde el principio para
80     //poder trabajar con él, el resto del programa
81     LecturaArchivoUDAS(UDA);
82
83     //aqui hacemos un menú con un do-while, en este menú están las diferentes
84     //opciones que tiene el usuario para realizar en el programa
85     do
86     {
87         printf("\n- Sistema de Registro de Calificaciones -\n");
88         printf("\n1. Ingresar Registro");
89         printf("\n2. Listado de Registros");
90         printf("\n3. Guardar Registros");
91         printf("\n4. Leer Registros");
92         printf("\n5. Sacar Promedio General");
93         printf("\n6. Sacar promedio por semestre");
94         printf("\n7. Sacar numero de creditos");
95         printf("\n8. Listado de las udas");
96         printf("\n9. Salir");
97         printf("\n\nOpcion: ");
98         //Se captura el resultado del usuario
99         scanf("%u",&opcion);
100        //Se abre un switch, para que, según la respeusta del usuario,

```

```

100 //Se abre un switch, para que, según la respuesta del usuario,
101 //se ejecute esa acción
102 switch(opcion)
103 {
104     case 1:
105         NumeroRegistros++;
106         CapturarRegistro(UDAcursada,NumeroRegistros);
107         break;
108
109     case 2:
110         for(k=0; k<NumeroRegistros; k++)
111             PrintRegistro(UDAcursada,k+1,UDA);
112         break;
113
114     case 3:
115         GuardarRegistros(UDAcursada,NumeroRegistros);
116         break;
117
118     case 4:
119         NumeroRegistros=LeerRegistros(UDAcursada);
120         break;
121
122     case 5:
123         //Se inicia un ciclo for que repasa los registros que se han hecho
124         for(k=0; k<NumeroRegistros; k++)
125             //aquí con la variable promedio, mandamos a llamar a la función y luego
126             //la imprimimos
127             promedio = (PromedioGeneral(UDAcursada,k+1,UDA))/NumeroRegistros;
128             printf("\n El promedio general es: %f\n",promedio);
129             break;
130
131     case 6:
132         break;
133
134     case 7:
135         //Se inciaa un ciclo for que repasa los registros que se han hecho
136         for(k=0; k<NumeroRegistros; k++)
137         {
138             //Con la variable aux, llamo a la función
139             aux = TotalCreditos(UDAcursada,k+1,UDA);
140             //voy sumando los resultados y los guardo en la variable tc
141             tc += aux;
142         }
143
144         //imprimo el total de creditos
145         printf("\nTotal de creditos: %u",tc);
146         break;
147
148     case 8:
149         listadou(UDA);
150         break;

```



```

151
152     case 9:
153         salir=1;
154         break;
155
156     default: printf("\nOpcion no valida.");
157 }
158 }
159 while(!salir);
160
161     printf(" \n");
162     return 0;
163 }
164
165 //Esta es una funcion para imprimir el nombre de las udas con el id y los creditos
166 void listadou(UNIDADdeAPRENDIZAJE UDA[])
167 {
168     FILE* Archivo;
169     unsigned int aux,k;
170     Archivo=fopen(NombreDelArhivoDeUDAS,"r");
171     if(Archivo!=NULL)
172     {
173         printf("\n Aquí puede buscar la UDA con su id, si es que no la sabe");
174         for(k=0; k<Total_UDAS; k++)
175         {
176             fscanf(Archivo,"%u",&aux);
177             UDA[k].Id=aux;
178             fgetc(Archivo);
179             fgets(UDA[k].Nombre,100,Archivo);
180             fgets(UDA[k].Area,50,Archivo);
181             fgets(UDA[k].Tipo,50,Archivo);
182             fscanf(Archivo,"%u",&aux);
183             UDA[k].NumCre=aux;
184             fscanf(Archivo,"%u",&aux);
185             UDA[k].HorasSem=aux;
186         }
187         fclose(Archivo);
188     }
189     else
190     {
191         printf("Error al leer el archivo ListadoUDA.txt o no se encuentra \n");
192         exit(0);
193     }
194     printf("\n\n");
195     for(k=0;k<Total_UDAS;k++)
196         Printuda(UDA[k]);
197
198     printf("\n");
199 }
200

```

```

200
201 //Se inicia la función para capturar los registros.
202 void CapturarRegistro(REGISTRO UDAcursada[], unsigned char NumReg)
203 {
204     //Se declaran las variables locales de esta función
205     unsigned char Indice;
206     unsigned int aux;
207     float Calif;
208
209
210     Indice=NumReg-1;
211     UDAcursada[Indice].Id=NumReg;
212
213     //Se le va preguntando al usuario todos los datos que quiere registrar
214     //para luego guardarlos
215     //Se pide el id de la uda, calificación y oportunidad en que se cursó.
216     printf("\nEscriba el Id correspondiente a la UDA cursada: ");
217     scanf("%u",&aux);
218     UDAcursada[Indice].Id_UDA=aux;
219     printf("Calificación: ");
220     scanf("%f",&Calif);
221     UDAcursada[Indice].Status.Calificacion=Calif;
222     printf("En que Oportunidad se curso: ");
223     scanf("%u",&aux);
224     UDAcursada[Indice].Status.Oportunidad=aux;
225
226     //Esta es alguna clase de if, pero comprimido por decirlo de alguna manera
227     //Regresa 2 valores, 0 o 1, false o true
228     UDAcursada[Indice].Status.Aprobada=(Calif<7.0)?0:1;
229
230     //Eliminamos la tecla "Enter" que se queda en el buffer del teclado despues de leer la oportunidad
231     //Se piden los datos del profesor que impartió la uda
232     gets(&UDAcursada[Indice].Profesor.Nombres[0]);
233     printf("Nombre(s) del profesor: ");
234     gets(&UDAcursada[Indice].Profesor.Nombres[0]);
235     printf("Apellido Paterno: ");
236     gets(&UDAcursada[Indice].Profesor.Apaterno[0]);
237     printf("Apellido Materno: ");
238     gets(&UDAcursada[Indice].Profesor.Amaterno[0]);
239 }
240
241 //Aqui se inicia la funcion para sacar el total de creditos
242 int TotalCreditos(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])
243 {
244     //decalramos una variable aux
245     unsigned int aux=0;
246
247     //abrimos un if, en el que si la materia está aprobada, aux vale la cantidad de creditos
248     if(UDAcursada[NumReg-1].Status.Aprobada)
249         aux = ((int)(ListadoUDAS[NumReg-1].NumCre));
250     //en caso de que la materia no haya sido aprobada, los creditos no se suman

```

```

250 //en caso de que la materia no haya sido aprobada, los creditos no se suman
251 else
252     aux = 0;
253
254     return aux;
255 }
256
257 //Aqui se inicia la funcion del promedio general
258 float PromedioGeneral(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])
259 {
260     //declaramos una variable float llamada aux
261     float aux;
262
263     // aqui se van sumando las calificaciones que ya registró el usuario
264     aux = aux + UDAcursada[NumReg-1].Status.Calificacion;
265
266     //se retorna la variable aux
267     return aux;
268 }
269
270 //Aqui se leen los registros hechos por el usuario
271 unsigned char LeerRegistros(REGISTRO UDAcursada[])
272 {
273     FILE* Archivo;
274     unsigned int aux,k;
275     unsigned char NumeroDeRegistros;
276
277     Archivo=fopen("registroUDAS.txt","r");
278     if(Archivo!=NULL)
279     {
280         printf("\n Se ha abierto el archivo");
281         fscanf(Archivo,"%u",&aux);
282         NumeroDeRegistros=aux;
283         //Aqui hace el recuento de los registros, al pasar por todos estos, aumenta k
284         //que es el numero de registros que se han hecho
285         for(k=0; k<NumeroDeRegistros; k++)
286         {
287             fscanf(Archivo,"%u",&aux);
288             UDAcursada[k].Id=aux;
289             fscanf(Archivo,"%u",&aux);
290             UDAcursada[k].Id_UDA=aux;
291             fscanf(Archivo,"%u",&UDAcursada[k].Status.Calificacion);
292             fscanf(Archivo,"%u",&aux);
293             UDAcursada[k].Status.Oportunidad=aux;
294             fscanf(Archivo,"%u",&aux);
295             UDAcursada[k].Status.Aprobada=aux;
296             fgetc(Archivo);
297             fgets(UDAcursada[k].Profesor.Nombres,40,Archivo);
298             fgets(UDAcursada[k].Profesor.Apaterno,20,Archivo);
299             fgets(UDAcursada[k].Profesor.Amaterno,20,Archivo);
300         }
301     }

```

```

300     }
301
302     fclose(Archivo);
303     printf("\n Se leyeron %u registros y se cerro el archivo\n",NumeroDeRegistros);
304
305     //al haber hecho el recuento, se imprime la cantidad de registros que se hicieron
306 }
307 else
308 {
309     printf("Error al leer el archivo ListadoUDA.txt o no se encuentra \n");
310     exit(0);
311 }
312 }
313
314 //Aquí se inicia la función para guardar los registros
315 void GuardarRegistros(REGISTRO UDAcursada[], unsigned int NumDeRegistros)
316 {
317     FILE* Archivo;
318     unsigned int i;
319
320     //Crea un archivo en modo escritura
321     Archivo=fopen("registroUDAS.txt","w");
322
323     //Validar el resultado de la creación del archivo
324     if(Archivo!=NULL)
325     {
326         //Se creo el archivo para escritura correctamente
327         fprintf(Archivo,"%u\n",NumDeRegistros);
328         for(i=0; i<NumDeRegistros; i++)
329         {
330             fprintf(Archivo,"%u\n",UDAcursada[i].Id);
331             fprintf(Archivo,"%u\n",UDAcursada[i].Id_UDA);
332             fprintf(Archivo,"%g\n",UDAcursada[i].Status.Calificacion);
333             fprintf(Archivo,"%u\n",UDAcursada[i].Status.Oportunidad);
334             fprintf(Archivo,"%u\n",UDAcursada[i].Status.Aprobada);
335             fprintf(Archivo,"%s\n",UDAcursada[i].Profesor.Nombres);
336             fprintf(Archivo,"%s\n",UDAcursada[i].Profesor.Apaterno);
337             fprintf(Archivo,"%s\n",UDAcursada[i].Profesor.Amaterno);
338         }
339         //Cerrar el archivo cuando ya terminamos de usarlo
340         fclose(Archivo);
341         printf("\nregistros guardados correctamente.\n");
342     }
343     else
344     //Error al crear el archivo para escritura
345     printf("Error: no se pudo crear el archivo \n");
346     printf(" \n");
347 }
348
349 //Esta funcion imprime los registros que el usuario haya ingresado
350 void PrintRegistro(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])

```

```

350 void PrintRegistro(REGISTRO UDAcursada[], unsigned char NumReg, UNIDADdeAPRENDIZAJE ListadoUDAS[])
351 {
352     //solo retoma los mismos datos que ingreso el usuario
353     //calificacion, oportunidad, profesores y si se que aprobo la materia
354     printf("\n Registro %i:",UDAcursada[NumReg-1].Id);
355     PrintUDA( ListadoUDAS[ UDAcursada[NumReg-1].Id_UDA - 1 ] );
356     printf("\n Calificacion: %2.2f",UDAcursada[NumReg-1].Status.Calificacion);
357     printf("\n Oportunidad: %i",UDAcursada[NumReg-1].Status.Oportunidad);
358     if(UDAcursada[NumReg-1].Status.Aprobada)
359         printf("\n UDA Aprobada");
360     else
361         printf("\n UDA NO Aprobada");
362     printf("\n Profesor: %s %s %s\n",UDAcursada[NumReg-1].Profesor.Nombres,UDAcursada[NumReg-1].Profesor.Apaterno,UDAcursada[NumReg-1].Profesor.Amaterno);
363 }
364
365 //Esta funcion es la que se encarga de sacar toda la informacion del archivo del que trabajamos
366 void LecturaArchivoUDAS(UNIDADdeAPRENDIZAJE UDA[])
367 {
368     FILE* Archivo;
369     unsigned int aux,k;
370     Archivo=fopen(NombreDelArchivoDeUDAS,"r"); //se abre el archivo
371     if(Archivo==NULL)
372     {
373         printf("\n Se ha abierto el archivo");
374         for(k=0; k<Total_UDAS; k++)
375         {
376             //de aqui extrae los datos que tiene el archivo y los guarda
377             fscanf(Archivo,"%u",&aux);
378             UDA[k].Id=aux;
379             fgets(Archivo);
380             fgets(UDA[k].Nombre,100,Archivo);
381             fgets(UDA[k].Area,50,Archivo);
382             fgets(UDA[k].Tipo,50,Archivo);
383             fscanf(Archivo,"%u",&aux);
384             UDA[k].NumCre=aux;
385             fscanf(Archivo,"%u",&aux);
386             UDA[k].HorasSem=aux;
387         }
388         fclose(Archivo); //se cierra el archivo
389         printf("\n Se leyeron todos los registros y se cerro el archivo");
390     }
391     else
392     {
393         printf("Error al leer el archivo ListadoUDA.txt o no se encuentra \n");
394         exit(0);
395     }
396 }
397
398 //esta funcion imprime todos los registros que hay en el archivo
399 void PrintUDA(UNIDADdeAPRENDIZAJE UDA)
400 {

```

```

400 {
401     printf("\nID: %u",UDA.Id);
402     printf("\nNombre: %s",UDA.Nombre);
403     printf("Area: %s",UDA.Area);
404     printf("Tipo: %s",UDA.Tipo);
405     printf("Numero de Creditos: %u",UDA.NumCre);
406     printf("\nHoras por Semana: %u",UDA.HorasSem);
407 }
408
409 //Esta funcion hace lo mismo que al anterior, pero solo imprime el nombre de la uda, el id y los creditos
410 void Printuda(UNIDADdeAPRENDIZAJE UDA)
411 {
412     printf("\nID: %u",UDA.Id);
413     printf("\nNombre: %s",UDA.Nombre);
414     printf("Numero de Creditos: %u\n",UDA.NumCre);
415 }

```

Pruebas y resultados

Evidencia de los códigos implementados

Usaré los ejemplos propuestos en la introducción de este reporte.

```
C:\Windows\system32\cmd.exe - p015

C:\Users\chama\Documents\algoritmos>gcc -c p015.c

C:\Users\chama\Documents\algoritmos>gcc p015.o -o p015.exe

C:\Users\chama\Documents\algoritmos>p015

Se ha abierto el archivo
Se leyeron todos los registros y se cerro el archivo
- Sistema de Registro de Calificaciones -

1. Ingresar Registro
2. Listado de Registros
3. Guardar Registros
4. Leer Registros
5. Sacar Promedio General
6. Sacar promedio por semestre
7. Sacar numero de creditos
8. Listado de las udas
9. Salir

Opcion: 1

Escriba el Id correspondiente a la UDA cursada: 1
Calificacion: 10
En que Oportunidad se curso: 1
Nombre(s) del profesor:  Javier
Apellido Paterno:  Hernandez
Apellido Materno:  Gazca
```

```
Opcion: 1

Escriba el Id correspondiente a la UDA cursada: 2
Calificacion: 7
En que Oportunidad se curso: 1
Nombre(s) del profesor:  Juan
Apellido Paterno:  Aguilera
Apellido Materno:  Huerta
```

Opcion: 1

Escriba el Id correspondiente a la UDA cursada: 3
Calificacion: 5
En que Oportunidad se curso: 1
Nombre(s) del profesor: Diego
Apellido Paterno: Perez
Apellido Materno: Perez

Escriba el Id correspondiente a la UDA cursada: 4
Calificacion: 8
En que Oportunidad se curso: 1
Nombre(s) del profesor: Jair
Apellido Paterno: Chavez
Apellido Materno: Islas

Escriba el Id correspondiente a la UDA cursada: 5
Calificacion: 9
En que Oportunidad se curso: 1
Nombre(s) del profesor: Jose
Apellido Paterno: Perez
Apellido Materno: Morales

Escriba el Id correspondiente a la UDA cursada: 6
Calificacion: 8
En que Oportunidad se curso: 1
Nombre(s) del profesor: Mariana
Apellido Paterno: Juarez
Apellido Materno: Perez

Entonces usando la el número 2, podemos ver todos los registros que hicimos

C:\> Seleccionar C:\Windows\system32\cmd.exe - p015

Opcion: 2

Registro 1:

ID: 1

Nombre: Actividades del Área Complementaria

Area: Complementaria

Tipo: Obligatoria

Numero de Creditos: 5

Horas por Semana: 0

Calificacion: 10.00

Oportunidad: 1

UDA Aprobada

Profesor: Javier Hernandez Gazca

Registro 2:

ID: 2

Nombre: Actividades del Área General

Area: General

Tipo: Obligatoria

Numero de Creditos: 8

Horas por Semana: 0

Calificacion: 7.00

Oportunidad: 1

UDA Aprobada

Profesor: Juan Aguilera Huerta

Registro 3:

ID: 3

Nombre: Administración de la Tecnología Informática

Area: Profundización

Tipo: Obligatoria

Numero de Creditos: 6

Horas por Semana: 4

Calificacion: 5.00

Oportunidad: 1

UDA NO Aprobada

Profesor: Diego Perez Perez

Registro 4:

ID: 4

Nombre: Administración de Proyecto de Tesis

Area: Complementaria

Tipo: Optativa Area Complementaria

Numero de Creditos: 3

Horas por Semana: 2

Calificacion: 8.00

Oportunidad: 1

UDA Aprobada

Profesor: Jair Chavez Islas

Registro 5:

ID: 5

Nombre: Álgebra Lineal

Area: Básica Común

Tipo: Obligatoria

Numero de Creditos: 6

Horas por Semana: 4

Calificacion: 9.00

Oportunidad: 1

UDA Aprobada

Profesor: Jose Perez Morales

Registro 6:

ID: 6

Nombre: Algoritmos de Procesamiento Digital de Señales

Area: Profundización

Tipo: Optativa Opción Informática Industrial

Numero de Creditos: 6

Horas por Semana: 4


```
C:\Windows\system32\cmd.exe - p015

- Sistema de Registro de Calificaciones -

1. Ingresar Registro
2. Listado de Registros
3. Guardar Registros
4. Leer Registros
5. Sacar Promedio General
6. Sacar promedio por semestre
7. Sacar numero de creditos
8. Listado de las udas
9. Salir

Opcion: 7

Total de creditos: 28
- Sistema de Registro de Calificaciones -
```

El número de créditos es el mismo que habíamos previsto

```
C:\Windows\system32\cmd.exe - p015

- Sistema de Registro de Calificaciones -

1. Ingresar Registro
2. Listado de Registros
3. Guardar Registros
4. Leer Registros
5. Sacar Promedio General
6. Sacar promedio por semestre
7. Sacar numero de creditos
8. Listado de las udas
9. Salir

Opcion: 5

El promedio general es: 7.833333
```

El promedio general da lo que habíamos previsto al principio

Con esto podemos concluir en que están bien las 2 funciones agregadas.

Cabe mencionar que me perdía porque no sabía cual eran los créditos de cada materia ni cual materia era ni su id, entonces agregué la función del número 8, que me imprime los id, nombre de las materias y sus créditos, nada más, de ahí pude sacar la información para saber si estaban correctos mis resultados

Opcion: 8

Aqui puede buscar la UDA con su id, si es que no la sabe

ID: 1

Nombre: Actividades del Área Complementaria

Numero de Creditos: 5

ID: 2

Nombre: Actividades del Área General

Numero de Creditos: 8

ID: 3

Nombre: Administración de la Tecnología Informática

Numero de Creditos: 6

ID: 4

Nombre: Administración de Proyecto de Tesis

Numero de Creditos: 3

ID: 5

Nombre: Álgebra Lineal

Numero de Creditos: 6

ID: 6

Nombre: Algoritmos de Procesamiento Digital de Señales

Numero de Creditos: 6

Esta semana estuve muy ocupado entre otras tareas y entre que fui a ponerme la vacuna y en lo que me recuperaba, así que eso dificultó un poco mi aprendizaje, pero aun así me llevé varias cosas como el manejo de archivos, el uso de las funciones, cómo es que puedo usar una función en otra y como puedo crear archivos nuevos con datos que yo incluya.