



Introducción a la Programación Practica 9

Medina Martinez Jonathan Jason 2023640061

12 de junio del 2023

Índice

1. Objetivo	3
2. Introducción	3
3. Desarrollo	4
3.1. Programa.c	5
3.1.1. Ejecución	10
4. Conclusión	15

1. Objetivo

Desarrollo de programas utilizando estructuras y archivos como aplicación para las bases de datos.

2. Introducción

El desarrollo de programas utilizando estructuras y archivos como aplicación para las bases de datos es una práctica fundamental en el ámbito de la programación. En esta ocasión, nos enfocaremos en crear un programa en C que permita gestionar una lista de asistencia de alumnos. Para lograr esto, utilizaremos diferentes estructuras, como Alumno, Profesor, Lista y Fecha, que contendrán la información relevante para el registro y seguimiento de la asistencia. Además, implementaremos un archivo binario que nos permitirá almacenar y acceder a los datos de manera aleatoria. Este enfoque nos brinda flexibilidad y eficiencia al momento de gestionar la información de los alumnos y su asistencia. A través de un menú de opciones, los usuarios podrán registrar unidades de aprendizaje, grupos, fechas, profesores, agregar y modificar alumnos, así como mostrar la lista de asistencia. Con esta práctica, adquiriremos conocimientos valiosos sobre el manejo de estructuras, archivos y bases de datos en el contexto de la programación.

3. Desarrollo

Desarrolle un programa en C que permita tener una lista de asistencia de alumnos. Para esto, deberá crear las siguientes estructuras:

- Alumno: Nombre, Numero de Boleta, Año de Ingreso, Carrera, Turno.
- Profesor: Nombre, Numero de Empleado, Turno.
- Lista: Fecha, Unidad de Aprendizaje, Grupo, Profesor, Lista de Alumnos.
- Fecha: Día, Mes, Año.

El programa deberá tener el siguiente menú:

- Registrar Unidad de Aprendizaje.
- Registrar Grupo.
- Registrar Fecha.
- Registrar Profesor.
- Agregar Alumno.
- Modificar Alumno.
- Mostrar Lista.
- Salir.

La Lista de Alumnos deberá guardarse en un archivo binario, que permita guardar y acceder a los datos de forma aleatoria. Para esto, podrá utilizar el numero de boleta como un identificador entero consecutivo. No olvide guardar en su archivo los datos del profesor, los datos de la lista y la cantidad de alumnos registrados.

3.1. Programa.c

```
1 /**
2  * @file Programa.c
3  * @author Medina Martinez Jonathan Jason (jmedinam1702@alumno.ipn.mx)
4  * @brief
5  * @version 0.1
6  * @date 2023-06-23
7  *
8  * @copyrigth GPLv3
9  *
10 */
11
12 #include <stdio.h>
13 #include <stdlib.h>
14
15 typedef struct {
16     int dia;
17     int mes;
18     int anio;
19 } Fecha;
20
21 typedef struct {
22     char nombre[50];
23     int numeroBoleta;
24     int anioIngreso;
25     char carrera[50];
26     char turno;
27 } Alumno;
28
29 typedef struct {
30     char nombre[50];
31     int numeroEmpleado;
32     char turno;
33 } Profesor;
34
35 typedef struct {
36     Fecha fecha;
37     char unidadAprendizaje[50];
38     char grupo[10];
39     Profesor profesor;
40     Alumno *alumnos;
41     int cantidadAlumnos;
42 } Lista;
43
```

```

44 void registrarUnidadAprendizaje(Lista *lista);
45 void registrarGrupo(Lista *lista);
46 void registrarFecha(Lista *lista);
47 void registrarProfesor(Lista *lista);
48 void agregarAlumno(Lista *lista);
49 void modificarAlumno(Lista *lista);
50 void mostrarLista(Lista lista);
51 void guardarDatos(Lista lista);
52 void cargarDatos(Lista *lista);
53
54 int main() {
55     Lista lista;
56     lista.alumnos = NULL;
57     lista.cantidadAlumnos = 0;
58
59     int opcion;
60
61     cargarDatos(&lista);
62
63     do {
64         printf("\n==== MENU ==== \n");
65         printf("1. Registrar Unidad de Aprendizaje \n");
66         printf("2. Registrar Grupo \n");
67         printf("3. Registrar Fecha \n");
68         printf("4. Registrar Profesor \n");
69         printf("5. Agregar Alumno \n");
70         printf("6. Modificar Alumno \n");
71         printf("7. Mostrar Lista \n");
72         printf("8. Salir \n");
73         printf("Ingrese una opcion: ");
74         scanf("%d", &opcion);
75
76         switch(opcion) {
77             case 1:
78                 registrarUnidadAprendizaje(&lista);
79                 break;
80             case 2:
81                 registrarGrupo(&lista);
82                 break;
83             case 3:
84                 registrarFecha(&lista);
85                 break;
86             case 4:
87                 registrarProfesor(&lista);

```

```

88         break;
89         case 5:
90             agregarAlumno(&lista);
91             break;
92         case 6:
93             modificarAlumno(&lista);
94             break;
95         case 7:
96             mostrarLista(lista);
97             break;
98         case 8:
99             guardarDatos(lista);
100             printf("Saliendo del programa...\n");
101             break;
102         default:
103             printf("Opcion invalida. Intente nuevamente.\n");
104     }
105     while(opcion != 8);
106
107     free(lista.alumnos);
108
109     return 0;
110 }
111
112 void registrarUnidadAprendizaje(Lista *lista) {
113     printf("\n==== REGISTRAR UNIDAD DE APRENDIZAJE ==== \n");
114     printf("Ingrese el nombre de la unidad de aprendizaje: ");
115     scanf("%s", lista->unidadAprendizaje);
116 }
117
118 void registrarGrupo(Lista *lista) {
119     printf("\n==== REGISTRAR GRUPO ==== \n");
120     printf("Ingrese el nombre del grupo: ");
121     scanf("%s", lista->grupo);
122 }
123
124 void registrarFecha(Lista *lista) {
125     printf("\n==== REGISTRAR FECHA ==== \n");
126     printf("Ingrese el dia: ");
127     scanf("%d", &lista->fecha.dia);
128     printf("Ingrese el mes: ");
129     scanf("%d", &lista->fecha.mes);
130     printf("Ingrese el anio: ");
131     scanf("%d", &lista->fecha.anio);

```

```

132 }
133
134 void registrarProfesor(Lista *lista) {
135     printf("\n=== REGISTRAR PROFESOR ===\n");
136     printf("Ingrese el nombre del profesor: ");
137     scanf("%s", lista->profesor.nombre);
138     printf("Ingrese el numero de empleado: ");
139     scanf("%d", &lista->profesor.numeroEmpleado);
140     printf("Ingrese el turno (M-maniana, T-tarde, N-noche): ");
141     scanf(" %c", &lista->profesor.turno);
142 }
143
144 void agregarAlumno(Lista *lista) {
145     printf("\n=== AGREGAR ALUMNO ===\n");
146
147     lista->cantidadAlumnos++;
148     lista->alumnos = (Alumno*)realloc(lista->alumnos, lista->cantidadAlumnos *
149                                     sizeof(Alumno));
150
151     Alumno *alumno = &lista->alumnos[lista->cantidadAlumnos - 1];
152
153     printf("Ingrese el nombre del alumno: ");
154     scanf("%s", alumno->nombre);
155     printf("Ingrese el numero de boleta: ");
156     scanf("%d", &alumno->numeroBoleta);
157     printf("Ingrese el anio de ingreso: ");
158     scanf("%d", &alumno->anioIngreso);
159     printf("Ingrese la carrera: ");
160     scanf("%s", alumno->carrera);
161     printf("Ingrese el turno (M-maniana, T-tarde, N-noche): ");
162     scanf(" %c", &alumno->turno);
163 }
164
165 void modificarAlumno(Lista *lista) {
166     printf("\n=== MODIFICAR ALUMNO ===\n");
167
168     int numeroBoleta;
169     printf("Ingrese el numero de boleta del alumno a modificar: ");
170     scanf("%d", &numeroBoleta);
171
172     int encontrado = 0;
173
174     for (int i = 0; i < lista->cantidadAlumnos; i++) {
175         if (lista->alumnos[i].numeroBoleta == numeroBoleta) {

```



```

175         printf("Ingrese el nuevo nombre del alumno: ");
176         scanf("%s", lista->alumnos[i].nombre);
177         printf("Ingrese el nuevo anio de ingreso: ");
178         scanf("%d", &lista->alumnos[i].anioIngreso);
179         printf("Ingrese la nueva carrera: ");
180         scanf("%s", lista->alumnos[i].carrera);
181         printf("Ingrese el nuevo turno (M-maniana, T-tarde, N-noche): ");
182         ;
183         scanf(" %c", &lista->alumnos[i].turno);
184
185         encontrado = 1;
186         break;
187     }
188 }
189
190 if (!encontrado) {
191     printf("Alumno no encontrado.\n");
192 }
193 }
194
195 void mostrarLista(Lista lista) {
196     printf("\n==== LISTA DE ASISTENCIA ==== \n");
197
198     printf("Fecha: %02d/%02d/%04d\n", lista.fecha.dia, lista.fecha.mes, lista.
199         fecha.anio);
200     printf("Unidad de Aprendizaje: %s\n", lista.unidadAprendizaje);
201     printf("Grupo: %s\n", lista.grupo);
202     printf("Profesor: %s\n", lista.profesor.nombre);
203     printf("Numero de empleado del profesor: %d\n", lista.profesor.numeroEmpleado);
204     ;
205     printf("Turno del profesor: %c\n", lista.profesor.turno);
206
207     printf("\nAlumnos:\n");
208     printf("%-20s %-15s %-10s %-10s\n", "Nombre", "Num. Boleta", "Anio Ingreso",
209         "Carrera");
210
211     for (int i = 0; i < lista.cantidadAlumnos; i++) {
212         Alumno alumno = lista.alumnos[i];
213         printf("%-20s %-15d %-10d %-10s\n", alumno.nombre, alumno.
214             numeroBoleta, alumno.anioIngreso, alumno.carrera);
215     }
216 }
217
218 void guardarDatos(Lista lista) {

```

```

214 FILE *archivo = fopen("lista_alumnos.bin", "wb");
215
216 if (archivo == NULL) {
217     printf("Error al abrir el archivo.\n");
218     return;
219 }
220
221 fwrite(&lista.fecha, sizeof(Fecha), 1, archivo);
222 fwrite(lista.unidadAprendizaje, sizeof(char), sizeof(lista.unidadAprendizaje),
223     archivo);
224 fwrite(lista.grupo, sizeof(char), sizeof(lista.grupo), archivo);
225 fwrite(&lista.profesor, sizeof(Profesor), 1, archivo);
226 fwrite(&lista.cantidadAlumnos, sizeof(int), 1, archivo);
227 fwrite(lista.alumnos, sizeof(Alumno), lista.cantidadAlumnos, archivo);
228
229 fclose(archivo);
230 }
231
232 void cargarDatos(Lista *lista) {
233     FILE *archivo = fopen("lista_alumnos.bin", "rb");
234
235     if (archivo == NULL) {
236         printf("No se encontro el archivo. Se creara uno nuevo.\n");
237         return;
238     }
239
240     fread(&lista->fecha, sizeof(Fecha), 1, archivo);
241     fread(lista->unidadAprendizaje, sizeof(char), sizeof(lista->unidadAprendizaje),
242         archivo);
243     fread(lista->grupo, sizeof(char), sizeof(lista->grupo), archivo);
244     fread(&lista->profesor, sizeof(Profesor), 1, archivo);
245     fread(&lista->cantidadAlumnos, sizeof(int), 1, archivo);
246
247     lista->alumnos = (Alumno*)malloc(lista->cantidadAlumnos * sizeof(Alumno));
248     fread(lista->alumnos, sizeof(Alumno), lista->cantidadAlumnos, archivo);
249
250     fclose(archivo);
251 }

```

3.1.1. Ejecución

```
1      No se encontro el archivo. Se creara uno nuevo.
2
3      ==== MENU ====
4      1. Registrar Unidad de Aprendizaje
5      2. Registrar Grupo
6      3. Registrar Fecha
7      4. Registrar Profesor
8      5. Agregar Alumno
9      6. Modificar Alumno
10     7. Mostrar Lista
11     8. Salir
12     Ingrese una opcion: 1
13
14     ==== REGISTRAR UNIDAD DE APRENDIZAJE ====
15     Ingrese el nombre de la unidad de aprendizaje: algebra
16
17     ==== MENU ====
18     1. Registrar Unidad de Aprendizaje
19     2. Registrar Grupo
20     3. Registrar Fecha
21     4. Registrar Profesor
22     5. Agregar Alumno
23     6. Modificar Alumno
24     7. Mostrar Lista
25     8. Salir
26     Ingrese una opcion: 2
27
28     ==== REGISTRAR GRUPO ====
29     Ingrese el nombre del grupo: 1mv3
30
31     ==== MENU ====
32     1. Registrar Unidad de Aprendizaje
33     2. Registrar Grupo
34     3. Registrar Fecha
35     4. Registrar Profesor
36     5. Agregar Alumno
37     6. Modificar Alumno
38     7. Mostrar Lista
39     8. Salir
40     Ingrese una opcion: 3
41
42     ==== REGISTRAR FECHA ====
43     Ingrese el dia: 12
```

```

44 Ingrese el mes: 06
45 Ingrese el anio: 2023
46
47 ==== MENU ====
48 1. Registrar Unidad de Aprendizaje
49 2. Registrar Grupo
50 3. Registrar Fecha
51 4. Registrar Profesor
52 5. Agregar Alumno
53 6. Modificar Alumno
54 7. Mostrar Lista
55 8. Salir
56 Ingrese una opcion: 4
57
58 ==== REGISTRAR PROFESOR ====
59 Ingrese el nombre del profesor: jose luis
60 Ingrese el numero de empleado: Ingrese el turno (M-maniana, T-tarde, N-noche):
61 ==== MENU ====
62 1. Registrar Unidad de Aprendizaje
63 2. Registrar Grupo
64 3. Registrar Fecha
65 4. Registrar Profesor
66 5. Agregar Alumno
67 6. Modificar Alumno
68 7. Mostrar Lista
69 8. Salir
70 Ingrese una opcion:
71 ==== REGISTRAR PROFESOR ====
72 Ingrese el nombre del profesor: Ingrese el numero de empleado: 12
73 Ingrese el turno (M-maniana, T-tarde, N-noche): T
74
75 ==== MENU ====
76 1. Registrar Unidad de Aprendizaje
77 2. Registrar Grupo
78 3. Registrar Fecha
79 4. Registrar Profesor
80 5. Agregar Alumno
81 6. Modificar Alumno
82 7. Mostrar Lista
83 8. Salir
84 Ingrese una opcion: 5
85
86 ==== AGREGAR ALUMNO ====
87 Ingrese el nombre del alumno: jonathan

```

```

88 Ingrese el numero de boleta: 2023640061
89 Ingrese el anio de ingreso: 2023
90 Ingrese la carrera: mecatronica
91 Ingrese el turno (M-maniana, T-tarde, N-noche): T
92
93 ==== MENU ====
94 1. Registrar Unidad de Aprendizaje
95 2. Registrar Grupo
96 3. Registrar Fecha
97 4. Registrar Profesor
98 5. Agregar Alumno
99 6. Modificar Alumno
100 7. Mostrar Lista
101 8. Salir
102 Ingrese una opcion: 6
103
104 ==== MODIFICAR ALUMNO ====
105 Ingrese el numero de boleta del alumno a modificar: 2023640061
106 Ingrese el nuevo nombre del alumno: jonathan
107 Ingrese el nuevo anio de ingreso: 2022
108 Ingrese la nueva carrera: mecatronica
109 Ingrese el nuevo turno (M-maniana, T-tarde, N-noche): M
110
111 ==== MENU ====
112 1. Registrar Unidad de Aprendizaje
113 2. Registrar Grupo
114 3. Registrar Fecha
115 4. Registrar Profesor
116 5. Agregar Alumno
117 6. Modificar Alumno
118 7. Mostrar Lista
119 8. Salir
120 Ingrese una opcion: 7
121
122 ==== LISTA DE ASISTENCIA ====
123 Fecha: 12/06/2023
124 Unidad de Aprendizaje: algebra
125 Grupo: 1mv3
126 Profesor: uis
127 Numero de empleado del profesor: 12
128 Turno del profesor: T
129
130 Alumnos:
131 Nombre Num. Boleta Anio Ingreso Carrera

```

```
132 jonathan 2023640061 2022 mecatronica
133
134 ==== MENU ====
135 1. Registrar Unidad de Aprendizaje
136 2. Registrar Grupo
137 3. Registrar Fecha
138 4. Registrar Profesor
139 5. Agregar Alumno
140 6. Modificar Alumno
141 7. Mostrar Lista
142 8. Salir
143 Ingrese una opcion: 8
144 Saliendo del programa...
```

4. Conclusión

En conclusión, el desarrollo de programas que utilicen estructuras y archivos como aplicación para las bases de datos es una habilidad fundamental en el ámbito de la programación. En esta práctica, hemos logrado crear un programa en C que nos permite gestionar una lista de asistencia de alumnos de manera eficiente y flexible. Mediante el uso de estructuras como Alumno, Profesor, Lista y Fecha, hemos sido capaces de almacenar y acceder a los datos relevantes para el registro y seguimiento de la asistencia. Además, gracias al archivo binario implementado, hemos logrado guardar y recuperar la información de forma aleatoria, aprovechando el número de boleto como identificador único. Esta práctica nos ha brindado la oportunidad de adquirir conocimientos valiosos sobre el manejo de estructuras, archivos y bases de datos en el contexto de la programación. Con estas habilidades, estaremos preparados para enfrentar desafíos más complejos en el desarrollo de aplicaciones que requieran gestionar grandes volúmenes de datos y garantizar un acceso eficiente a los mismos.