


| | | |
|---|-------------------------|--|
|  | <h2>Proyecto Final</h2> | |
| Facultad de Ingeniería | Laboratorio de docencia | |

Profesor: MARCO ANTONIO MARTINEZ QUINTANA

Asignatura: ESTRUCTURA DE DATOS Y ALGORITMOS I

Grupo: 17

Proyecto: BASE DE DATOS

Alumno: José Luis Arroyo Chavarría

No. de Lista : 5

Semestre: 2

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

Objetivo:

Tener una base de guardado de una gran variedad de documentos y con su clasificación de estos dependiendo de hora, fecha y/o Tema

Alcance de su proyecto:

Esto en el futuro se puede utilizar para las compañías, departamentos de gobierno (Secretarías) o uso para cualquier persona para que estén programados en días y horas que avisaran al usuario cuando van a alcanzar su límite de entrega si es en caso de las compañías(cronometro o alarma), además con una implementación antihackeo. Pero esto se agregara a futuro y dependiendo de lo aprendido durante mi carrera.

Introducción:

Durante lo visto en la materia de Estructuras y Datos 1 hemos desarrollado u observado la forma de almacenamiento de datos que han utilizado compañías como Facebook, Windows y en lo que me interesa los videojuegos. En este caso me intereso mucho el hacer este proyecto.

Al principio al platicar con amigos sobre lo que sería nuestros proyectos este empezó como una pokedex (enciclopedia de Pokemones) ya que en la aplicación llama Pokemon Go no se tiene una descripción y/o evoluciones de estas criaturas a como los tiene los juegos así nombrados, en este caso seria los primeros 150.

Pero este cambio al platicar con mi madre que trabaja en la Policía Federal actualmente Guardia Nacional en donde me estaba diciendo que la manera de ordenar documentos o su base de datos es por medio de la aplicación de Excel en donde está clasificado por fecha, asunto y documentos. En esta forma de uso para un departamento de gobierno o compañía es muy obsoleta para esta ya en un caso de emergencia seria de una forma muy lenta y obsoleta a lo que nuestra tecnología nos puede permitir.

Al final de lo platicado y reflexionado he decido hacer un programa en donde se podrá manejar por medio de apartados en donde se clasificara de lo visto o lo que desee el usuario en donde dependiendo de sus usos.

Desarrollo:

Lenguaje a utilizar: C

En este proyecto se utilizara varios temas vistos en nuestras clases como los arreglos y los apuntadores pero en lo visto en lo que se va a realizar se ocupara precisamente los datos abstractos, esto nos servirá para la clasificación de datos o documentos y más cuando será cuando se agregue o se tenga que ver próximamente.

- **Definición:**

- ❖ **Arreglo:**

Es un conjunto de datos finito y del mismo tipo. En realidad funciona como cualquier variable cualquiera, excepto que en lugar de almacenar un solo valor, guarda algunos valores. Pueden ser unidimensionales o multidimensionales. Los arreglos nos permiten hacer un conjunto de operaciones para manipular los datos guardados en ellos, estas operaciones son: ordenar, buscar, insertar, eliminar, modificar entre otras.

- ❖ **Apuntador:**

Es una variable que contiene una dirección de memoria, la cual corresponderá a un dato o a una variable que contiene el dato. Cada variable que se utiliza en una aplicación ocupa una o varias posiciones de memoria. Estas posiciones de memoria se accedan por medio de una dirección

- ❖ **Datos abstractos:**

Un tipo de dato abstracto (TDA) es un conjunto de datos u objetos creado de manera personalizada por un programador para un fin específico. Un TDA es una abstracción que permite modelar las características de un elemento en particular.

Un tipo de dato abstracto se puede manipular de forma similar a los tipos de datos que están predefinidos dentro del lenguaje de programación, encapsulando más información, según se requiera.

La implementación de un tipo de dato abstracto depende directamente del lenguaje de programación que se utilice. En lenguaje C los tipos de dato abstracto se crean mediante las estructuras (struct).

- **Algoritmo:**

Entradas:

- Registro de archivos para que el usuario desee poner
- Modificar, mostrar, buscar y eliminar los registros guardados

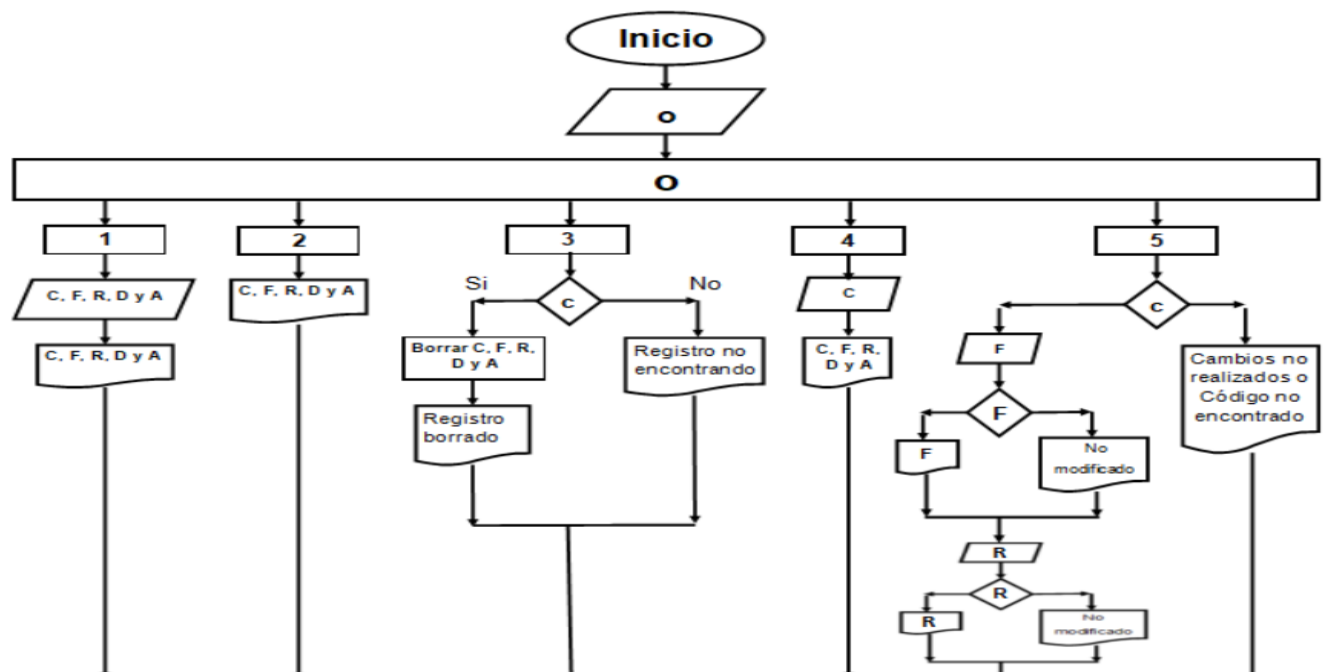
Salida:

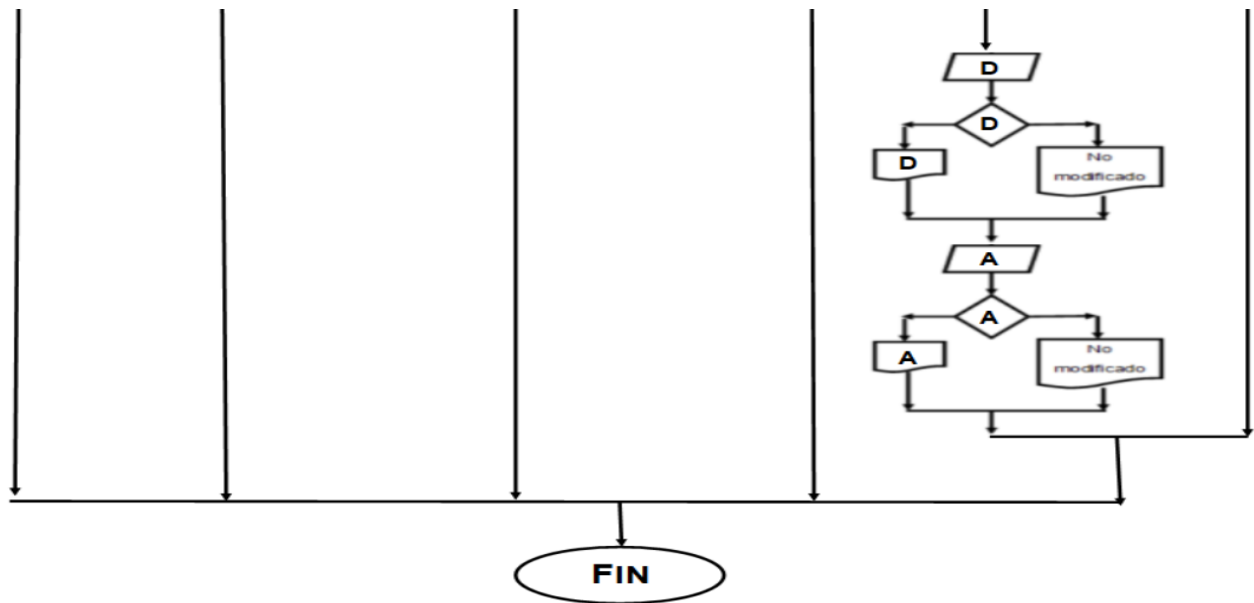
- Base de datos donde están los registros guardados que el usuario podrá utilizar dependiendo lo que desee.

Proceso:

- Ubicar por medio de un código los archivos a guardar y hacer procesos para ejecutar las acciones que podrá el usuario utilizar.

- **Diagrama de flujo:**





• **Pseudocódigo:**

| Variable: | Descripción: | Tipo: |
|-----------|--------------------------|-------------|
| O | Opción a elegir | Real |
| C | Código | Caracteres |
| F | Fecha o numero | Entero |
| R | Nombre del Registro | Información |
| D | Descripción del registro | Información |
| A | Archivo | Información |

1. Inicio
2. Leer O
3. Según O hacer
4. Caso 1
5. Leer C, F, R, D y A
6. Escribir C, F, R, D y A
7. Caso 2
8. Escribir C, F, R, D y A
9. Caso 3
10. Leer C
11. Si C existe entonces

12. Borrar C, F, R, D y A
13. Escribir Registro borrado
14. Sino
15. Escribir Registro no encontrando
16. FinSi
17. Caso 4
18. Leer C
19. Escribir C, F, R, D y A
20. Caso 5
21. Leer C
22. Si C existe entonces
23. Leer F
24. Si F entonces
25. Escribir F
26. Sino
27. Escribir Fecha o número no modificado
28. FinSi
29. Leer R
30. Si R entonces
31. Escribir R
32. Sino
33. Escribir Nombre del registro no modificado
34. FinSi
35. Leer D
36. Si D entonces
37. Escribir D
38. Sino
39. Escribir Descripción no modificado
40. FinSi
41. Leer A
42. Si A entonces

43. Escribir A
44. Sino
45. Escribir Archivo no modificado
46. FinSi
47. Sino
48. Escribir Cambios no realizados o Código no encontrado
49. FinSi
50. FinSegun
51. Fin

- **Código**

```
1 #include <stdio.h> /* Funciones principales */
2 #include <string.h> /* operaciones de manipulación de memoria */
3 #include <stdlib.h> /* Gestión de memoria dinámica, control de procesos, etc */
4 #include <locale.h> /* configuración regional actual */
5
6 #define MAX 1
7 #define VALOR_CENTINELA -1
8
9 struct informacion {
10     int codigo;
11     char nombre[MAX];
12     char descripcion[MAX];
13     char archivo[MAX];
14 };
15
16 typedef struct informacion Informacion;
17
18 void menuPrincipal();
19 void menuInsertar();
20 void menuBuscar();
21 void menuEliminar();
22 void menuMostrar();
23 void menuModificar();
24 void menuEliminarFisica();
25
26 /* Funciones para manejar el archivo directamente */
27 Informacion *obtenerInformaciones(int *n); /* Obtiene un vector dinámico de archivos */
28 char existeInformacion(int codigoInformacion, Informacion *informacion);
29 char insertarInformacion(Informacion informacion);
30 char eliminarInformacion(int codigoInformacion);
31 char eliminacionFisica();
32 char modificarInformacion(Informacion informacion);
33 char guardarReporte(); /* Genera un archivo TXT */
34
35 /* Función de lectura de cadenas */
36 int leecad(char *cad, int n);
37
38 /* Titular del programa */
39
40 void tituloPrincipal();
41
42 char linea[MAX];
43
44 int main()
45 {
46     setlocale(LC_ALL, "spanish"); /* Permite imprimir caracteres con tilde */
47     menuPrincipal();
48
49     return 0;
50 }
51
52 void menuPrincipal()
53 {
54     char repite = 1;
55     int opcion = -1;
56     /* Cuando el usuario ingresa texto en lugar de ingresar una opción. El programa no modifica
57     el valor de opcion. En ese caso, no se debe de ingresar a ninguno de los case, por eso se está
58     inicializando la variable opcion con un valor que no permita ejecutar ningún case. Simplemente,
59     volver a interactuar y pedir nuevamente la opción. */
60
61     do {
62         system("cls");
63
64         tituloPrincipal();
65
66         printf("\n\t\t\t\t\tMENU PRINCIPAL\n");
67         printf("\n\t\t\t\t\t1. Insertar nuevo registro\n");
68         printf("\t\t\t\t\t2. Mostrar listado de registros\n");
69         printf("\t\t\t\t\t3. Eliminar un registro\n");
70         printf("\t\t\t\t\t4. Buscar registro por numero o fecha\n");
71         printf("\t\t\t\t\t5. Modificar un registro\n");
72         printf("\t\t\t\t\t6. Eliminación física de registros\n");
73         printf("\t\t\t\t\t7. Salir\n");
74         printf("\n\t\t\t\t\tIngrese su opción: [ ]\b\b");
75
76         /* Lectura segura de un entero */
77         leecad(linea, MAX);
78         sscanf(linea, "%d", &opcion);
```

```

80         switch (opcion) {
81
82             case 1:
83                 menuInsertar();
84                 break;
85
86             case 2:
87                 menuMostrar();
88                 break;
89
90             case 3:
91                 menuEliminar();
92                 break;
93
94             case 4:
95                 menuBuscar();
96                 break;
97
98             case 5:
99                 menuModificar();
100                 break;
101
102             case 6:
103                 menuEliminarFisica();
104                 break;
105
106             case 7:
107                 repite = 0;
108                 break;
109         }
110     } while (repite);
111 }
112
113 void menuInsertar(){
114     Informacion informacion;
115     int codigoInformacion = 0;
116     char repite = 1;
117
118     char respuesta[MAX];
119
120     do {
121         system("cls");
122         tituloPrincipal();
123         printf("\n\t\t\t\t==> INSERTAR INFORMACION <==\n");
124
125         /* Se pide el código del producto a insertar */
126         printf("\n\tFecha o numero: ");
127         leecad(linea, MAX);
128         sscanf(linea, "%s", &codigoInformacion);
129
130         /* Se verifica que el producto no haya sido almacenado anteriormente */
131         if (!lexisteInformacion(codigoInformacion, &informacion)){
132
133             informacion.codigo = codigoInformacion;
134
135             /* Se piden los demás datos del producto a insertar */
136             printf("\tNombre del registro: ");
137             leecad(informacion.nombre, MAX);
138
139             printf("\tDescripcion: ");
140             leecad(informacion.descripcion, MAX);
141
142             printf("\tArchivo: ");
143             leecad(informacion.archivo, MAX);
144
145             if (insertarInformacion(informacion)) {
146                 printf("\n\tEl registro fue insertado correctamente\n");
147             } else {
148                 printf("\n\tOcurrió un error al intentar insertar el registro\n");
149                 printf("\tInténtelo mas tarde\n");
150             }
151         } else {
152             /* El registro ya existe, no puede ser insertado. */
153             printf("\n\tLa información de fecha o numero %f ya existe.\n", codigoInformacion);
154             printf("\tNo puede ingresar dos registros distintos con el mismo código.\n");
155         }
156     }
157
158     printf("\n\tDesea seguir ingresando registros? [S/N]: ");
159     leecad(respuesta, MAX);
160
161     if (!(strcmp(respuesta, "S") == 0 || strcmp(respuesta, "s") == 0)) {
162         repite = 0;
163     }
164
165     } while (repite);
166 }
167
168 void menuBuscar(){
169     Informacion informacion;
170     int codigoInformacion;
171     char repite = 1;
172     char respuesta[MAX];
173
174     do {
175         system("cls");
176         tituloPrincipal();
177         printf("\n\t\t\t\t==> BUSCAR REGISTRO POR FECHA O NUMERO <==\n");
178
179         /* Se pide el código del producto a buscar */
180         printf("\n\tFecha o numero del registro: ");
181         leecad(linea, MAX);
182         sscanf(linea, "%s", &codigoInformacion);
183
184         /* Se verifica que el registro a buscar, exista */
185         if (existeInformacion(codigoInformacion, &informacion)) {
186
187             /* Se muestran los datos del producto */
188             printf("\n\tFecha o numero del registro: %s\n", informacion.codigo);
189             printf("\tNombre del registro: %s\n", informacion.nombre);
190             printf("\tDescripcion del registro: %s\n", informacion.descripcion);
191             printf("\tArchivo del registro: %s\n", informacion.archivo);
192
193         } else {
194             /* El producto no existe */
195             printf("\n\tEl registro de la fecha o numero %s no existe.\n", codigoInformacion);
196         }
197     }
198
199     printf("\n\tDesea seguir buscando algún producto? [S/N]: ");
200     leecad(respuesta, MAX);
201
202     if (!(strcmp(respuesta, "S") == 0 || strcmp(respuesta, "s") == 0)) {
203         repite = 0;
204     }
205
206     } while (repite);
207 }
208
209 void menuEliminar()
210 {
211     Informacion informacion;
212     int codigoInformacion;
213     char repite = 1;
214     char respuesta[MAX];
215
216     do {
217         system("cls");
218         tituloPrincipal();
219         printf("\n\t\t\t\t==> ELIMINAR REGISTRO POR FECHA O NUMERO <==\n");
220
221         /* Se pide la fecha o el numero del registro a eliminar */
222         printf("\n\tFecha o numero del registro: ");
223         leecad(linea, MAX);
224         sscanf(linea, "%s", &codigoInformacion);
225
226         /* Se verifica que el registro a buscar, exista */
227         if (existeInformacion(codigoInformacion, &informacion)) {
228
229             /* Se muestran los datos del registro */
230             printf("\n\tFecha o numero del registro: %d\n", informacion.codigo);
231             printf("\tNombre del registro: %s\n", informacion.nombre);
232             printf("\tDescripcion del registro: %s\n", informacion.descripcion);
233             printf("\tArchivo del registro: %s\n", informacion.archivo);
234
235             printf("\n\tSeguro que desea eliminar el registro? [S/N]: ");
236             leecad(respuesta, MAX);
237             if (strcmp(respuesta, "S") == 0 || strcmp(respuesta, "s") == 0) {

```



```

395  /* Se pide el código del registro a eliminar */
396  printf("\n¿Seguro que desea proceder con la eliminación física? [S/N]: ");
397  lecad(respuesta, MAX);
398
399  if (strcmp(respuesta, "S") == 0 || strcmp(respuesta, "s") == 0) {
400      if (eliminarInformacion()) {
401          printf("\nLa eliminación física se realizó con éxito.\n");
402      } else {
403          printf("\nOcurrió algún error en la eliminación física.\n");
404      }
405  }
406
407  system("pause\n");
408  }
409
410  Informacion *obtenerInformacion(int *n){
411
412      FILE *archivo;
413      Informacion informacion;
414      Informacion *informaciones; /* Vector dinámico de informacion */
415      int i;
416
417      /* Abre el archivo en modo lectura */
418      archivo = fopen("informacion.dat", "rb");
419
420      if (archivo == NULL) { /* Si no se pudo abrir el archivo, el valor de archivo es NULL */
421          *n = 0; /* No se pudo abrir. Se considera n = 0 */
422          informaciones = NULL;
423      } else {
424
425          fseek(archivo, 0, SEEK_END); /* Posiciona el cursor al final del archivo */
426          *n = ftell(archivo) / sizeof(informacion); /* # de registros almacenados en el archivo. (# de registros) */
427          informaciones = (Informacion *)malloc(*n * sizeof(informacion)); /* Se reserva memoria para todos los registros almacenados en el */
428
429          /* Se recorre el archivo secuencialmente */
430          fseek(archivo, 0, SEEK_SET); /* Posiciona el cursor al principio del archivo */
431          fread(&informacion, sizeof(informacion), 1, archivo);
432          i = 0;
433
434          while (!feof(archivo)) {
435              informaciones[i++] = informacion;
436              fread(&informacion, sizeof(informacion), 1, archivo);
437          }
438
439          /* Cierra el archivo */
440          fclose(archivo);
441      }
442
443      return informaciones;
444  }
445
446  char existeInformacion(int codigoInformacion, Informacion *informacion)
447  {
448      FILE *archivo;
449      char existe;
450
451      /* Abre el archivo en modo lectura */
452      archivo = fopen("informacion.dat", "rb");
453
454      if (archivo == NULL) { /* Si no se pudo abrir el archivo, el valor de archivo es NULL */
455          existe = 0;
456      } else {
457          existe = 0;
458
459          /* Se busca el registro cuyo código coincida con codigoInformacion */
460          fread(&informacion, sizeof(*informacion), 1, archivo);
461          while (!feof(archivo)) {
462              if ((*informacion).codigo == codigoInformacion) {
463                  existe = 1;
464                  break;
465              }
466              fread(&informacion, sizeof(*informacion), 1, archivo);
467          }
468
469          /* Cierra el archivo */
470          fclose(archivo);
471      }
472  }
473
474      return existe;
475  }
476
477  char insertarInformacion(Informacion informacion)
478  {
479      FILE *archivo;
480      char insercion;
481
482      /* Abre el archivo para agregar datos al final */
483      archivo = fopen("informacion.dat", "ab"); /* Añade datos al final. Si el archivo no existe, es creado */
484
485      if (archivo == NULL) { /* Si no se pudo abrir el archivo, el valor de archivo es NULL */
486          insercion = 0;
487      } else {
488          fwrite(&informacion, sizeof(informacion), 1, archivo);
489          insercion = 1;
490
491          /* Cierra el archivo */
492          fclose(archivo);
493      }
494
495      return insercion;
496  }
497
498  /* Eliminación lógica de un registro */
499  char eliminarInformacion(int codigoInformacion)
500  {
501      FILE *archivo;
502      FILE *auxiliar;
503      Informacion informacion;
504      char elimina;
505
506      /* Abre el archivo para leer */
507      archivo = fopen("informacion.dat", "rb"); /* Modo lectura/escritura. Si el archivo no existe, es creado */
508
509      if (archivo == NULL) { /* Si no se pudo abrir el archivo, el valor de archivo es NULL */
510          elimina = 0;
511      }
512
513      while (!feof(archivo)) {
514          /* Se busca el registro que se quiere borrar. Cuando se encuentra, se sitúa en esa posición mediante la
515             función fseek y luego se modifica el campo clave de ese registro mediante algún valor centinela, eso se logra
516             con fwrite. Hasta allí se ha logrado una eliminación LÓGICA. Porque el registro sigue ocupando espacio en el archivo físico */
517
518          elimina = 0;
519          fread(&informacion, sizeof(informacion), 1, archivo);
520          while (!feof(archivo)) {
521              if (informacion.codigo == codigoInformacion) {
522                  fseek(archivo, ftell(archivo) - sizeof(informacion), SEEK_SET);
523                  informacion.codigo = VALOR_CENTINELA;
524                  fwrite(&informacion, sizeof(informacion), 1, archivo);
525                  elimina = 1;
526                  break;
527              }
528              fread(&informacion, sizeof(informacion), 1, archivo);
529          }
530
531          /* Cierra el archivo */
532          fclose(archivo);
533      }
534
535      return elimina;
536  }
537
538  char eliminacionFisica()
539  {
540      FILE *archivo;
541      FILE *temporal;
542      Informacion informacion;
543      char elimina = 0;
544
545      archivo = fopen("informacion.dat", "rb");
546      temporal = fopen("temporal.dat", "wb");
547
548      if (archivo == NULL || temporal == NULL) {
549          elimina = 0;
550      } else {
551          /* Se copia en el archivo temporal los registros válidos */

```


Resultados:

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

MENU PRINCIPAL

[1]. Insertar nuevo registro
[2]. Mostrar listado de registros
[3]. Eliminar un registro
[4]. Buscar registro por numero o fecha
[5]. Modificar un registro
[6]. Eliminación física de registros
[7]. Salir

Ingrese su opción: [1]
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> INSERTAR INFORMACION <==

Codigo: 652020

Fecha o numero: 06/05/2020
Nombre del registro: Registro de Prueba
Descripcion: Ejemplo
Archivo: Prueba.pdf

El registro fue insertado correctamente

Desea seguir ingresando registros? [S/N]: N
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> LISTADO DE REGISTROS INGRESADOS <==

-----
CODIGO      FECHA O NUMERO      REGISTRO      DESCRIPCION      ARCHIVO
-----
652020      06/05/2020          Registro de Prueba  Ejemplo          Prueba.pdf

Desea guardar el reporte en un archivo de texto? [S/N]: S

El reporte fue guardado con éxito
```

reporte.txt: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
==> LISTADO DE DATOS REGISTRADOS <==

-----
CODIGO      FECHA O NUMERO      REGISTRO      DESCRIPCION      ARCHIVO
-----
652020      06/05/2020          Registro de Prueba  Ejemplo          Prueba.pdf
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> BUSCAR REGISTRO POR CODIGO <==

Codigo del registro: 652020

Codigo del registro: 652020

Fecha o numero del registro: 06/05/2020
Nombre del registro: Registro de Prueba
Descripcion del registro: Ejemplo
Archivo del registro: Prueba.pdf

Desea seguir buscando algún producto? [S/N]:
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> MODIFICAR REGISTRO POR CODIGO <==

Codigo del registro: 652020

Fecha o numero del registro: 06/05/2020

Nombre del registro: Registro de Prueba
Descripcion: Ejemplo
Archivo: Prueba.pdf

Elija los datos a modificar

Fecha o numero: 06/05/2020
Desea modificar la fecha o numero? [S/N]: N

Nombre del registro actual: Registro de Prueba
Desea modificar el nombre del registro? [S/N]: S
Nuevo nombre del registro: Hola

Descripcion: Ejemplo
Desea modificar la descripcion? [S/N]: s
Nueva descripcion del registro: Adios

Archivo del registro actual: Prueba.pdf
Desea modificar el archivo del registro? [S/N]: n

Está seguro que desea modificar los datos del registro? [S/N]: s

El registro fue modificado correctamente

Desea modificar algún otro registro? [S/N]:
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                          PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> LISTADO DE REGISTROS INGRESADOS <==

-----
CODIGO      FECHA O NUMERO      REGISTRO      DESCRIPCION      ARCHIVO
-----
652020      06/05/2020         Hola          Adios             Prueba.pdf

Desea guardar el reporte en un archivo de texto? [S/N]:
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                    PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> ELIMINAR REGISTRO POR FECHA O NUMERO <==

Codigo del registro: 652020
Codigo del registro: 652020
Fecha o numero del registro: 06/05/2020
Nombre del registro: Hola
Descripcion del registro: Adios
Archivo del registro: Prueba.pdf

Seguro que desea eliminar el registro? [S/N]: S

Registro eliminado satisfactoriamente.

Desea eliminar otro registro? [S/N]: N
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                    PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> LISTADO DE REGISTROS INGRESADOS <==
-----
CODIGO      FECHA O NUMERO      REGISTRO      DESCRIPCION      ARCHIVO
-----
Desea guardar el reporte en un archivo de texto? [S/N]: _
```

C:\Users\luisw\Desktop\tareas UNAM\segundo semestre\datos y estructuras 1\Proyecto final\Proyecto final .exe

```
=====
                    PROYECTO FINAL
Base de datos: Creación, reportes, eliminación, búsqueda y actualización
Arroyo Chavarria Jose Luis
=====

==> ELIMINAR FÍSICAMENTE REGISTROS DEL ARCHIVO <==

Seguro que desea proceder con la eliminación física? [S/N]: S

La eliminación física se realizó con éxito.
```

Conclusiones:

Durante el desarrollo de mi proyecto tuve algunas dificultades pero con ayuda de mis apuntes de referencias de internet o de la facultad pude resolver mis problemas que había en el desarrollo. En parte final del proyecto me agrado la conclusión de mi proyecto aunque es solo un prototipo puede llegar a mas hasta un programa que tanto empresas o personas pueda utilizar y con el aprendizaje que este obteniendo se tendrá un mejor desempeño y mejor diseño.

Referencias:

- [https://es.wikipedia.org/wiki/Memoria_din%C3%A1mica_\(programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Memoria_din%C3%A1mica_(programaci%C3%B3n))
- <http://www.utn.edu.ec/reduca/programacion/arreglos/definiciones1.html>
- [http://www.utm.mx/~mgarcia/PE7\(Apuntadores\).pdf](http://www.utm.mx/~mgarcia/PE7(Apuntadores).pdf)
- Guía práctica de estudio 03. Manual de prácticas del Laboratorio de Estructuras de datos y algoritmos I. Tipo de dato abstracto. M.C. Edgar E. García Cano e Ing. Jorge A. Solano Gálvez. UNAM. 2017. Pags. 25 - 32
- <https://algoritmosyalgomas.com/archivos-binarios-en-c-insertar-eliminar-actualizar-listar/>
- http://gonzalopastor.ar.tripod.com/lenguaje_c/manual_bulanti/cursoc22.html
- <https://www.youtube.com/watch?v=8wKRS9TGplQ>
- <http://telematico-tools.azurewebsites.net/>
- <https://www.youtube.com/watch?v=uS6Ol2Xgps>