



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Antonio Ramos Gonzalez

Matrícula: 372576

Maestro: Pedro Núñez Yépiz

Actividad No. : 13

Tema - Unidad : Archivos

Ensenada Baja California a 19 de noviembre del 2023



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. INTRODUCCIÓN

La gestión de archivos es una parte fundamental en lo que respecta al almacenamiento de datos. Dichos archivos nos sirven para guardar todo tipo de registros y poder manipularlos cuantas veces queramos.

2. COMPETENCIA

En esta práctica se busca que el alumno comprenda de manera sólida la manipulación de archivos de texto, logrando leer y escribir en dichos archivos. También se busca reforzar temas vistos en prácticas anteriores como lo son los métodos de ordenación y búsqueda, validaciones, uso de funciones y el uso de nuestra librería propia

3. FUNDAMENTOS

Los archivos binarios son un tipo de archivo que su contenido este encriptado de forma que no se puede modificar de forma externa.

Para acceder a ellos se hace de la misma manera que los archivos de texto

```
File *ap;
```

```
Ap=fopen("Nombre.dll", "rb");
```

A la hora de querer leer o escribir dentro de un archivo binario se utilizará

```
Treg temp;
```

```
fread(&temp, sizeof(Treg), 1, ap);
```

```
fwrite(&temp, sizeof(Treg), 1, ap);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

4. PROCEDIMIENTO

MENU

- 1.- AGREGAR (AUTOM 100 REGISTROS)
- 2.- EDITAR REGISTRO
- 3.- ELIMINAR REGISTRO (lógico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6- IMPRIMIR
- 7.- GENERAR ARCHIVO TEXTO
- 8.- VER ARCHIVO TEXTO
- 9.- CREAR ARCH BINARIO
- 10.- CARGAR ARCH BINARIO
- 11.- MOSTRAR ELIMINADOS
- 0.- SALIR

UTILIZAR UN ARREGLO DE 5000 REGISTROS

SE DEBERÁ UTILIZAR ESTRUCTURAS CON LOS DATOS BÁSICOS DE UN EMPLEADO

preguntar nombre de archivo binario o de archivo texto

Busqueda y Ordenacion por CAMPO LLAVE

nota: usar librería propia con funciones

nota2: 100 % validado, Cuidar desbordamiento de vector

nota3: Campo llave matricula no repetido, archivos solo cargar 1 sola vez.

nota4: Usar el tipo Tkey para hacer mas practico el programa



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. RESULTADOS Y CONCLUSIONES

Los archivos binarios un tipo de archivo indispensable para el almacenamiento de datos importantes, debido a que a pesar de que se pueda acceder a ese archivo, no se puede cambiar u contenido de ninguna manera.

```
1 // Antonio Ramos Gonzalez Mt: 372576
2 // 14/11/2023 || 19/11/2023
3 // Se generara un archivo donde se almacenara datos generados por el usuario, y la lectura de archivos
4 // RGA_Act13_932
5 #include "Babilonia.h"
6 #define N 5000
7 > typedef struct _Reg --
8 } treg;
9
10 int msg();
11 void menu();
12 int verMt(treg reg[], int n, int mt);
13 treg agregar(treg reg[], int i);
14 void nombreAl(char nombre[], int sex);
15 void puestoAl(char puesto[]);
16 void apAl(char apellido[]);
17 int searchSec(treg reg[], int n, int mt);
18 int searchBin(treg reg[], int inf, int sup, int mt);
19 int orderBu(treg reg[], int n);
20 void pintReg(treg reg[], int n);
21 void pintOneReg(treg reg[], int n);
22 void swap(treg reg[], int i, int j);
23 void createTXT(treg reg[], int n);
24 int partition(treg reg[], int low, int high);
25 void quicksort(treg reg[], int low, int high);
26 void regEliminados(treg reg[], int n);
27 void modReg(treg reg[], int n);
28 void MostrarArchivo();
29 void archivoBin(treg reg[], int n);
30 int cargarArchBin(treg reg[], int n);
31 void BinArchEl();
32
33 > int main() --
34
35 > int msg()--
36
37 > void menu()--
38
39 > void nombreAl(char nombre[], int sex) --
40
41 > void apAl(char apellido[]) --
42
43 > treg agregar(treg reg[], int i) --
44
45 > void puestoAl(char puesto[]) --
46
47 > int searchSec(treg reg[], int n, int mt) --
48
49 > int searchBin(treg reg[], int inf, int sup, int mt) --
50
51 > int orderBu(treg reg[], int n) --
52
53 > void pintReg(treg reg[], int n) --
54
55 > void pintOneReg(treg reg[], int n) --
56
57 > int verMt(treg reg[], int n, int mt) --
58
59 > void createTXT(treg reg[], int n) --
60
61 > void swap(treg reg[], int i, int j) --
62
63 > int partition(treg reg[], int low, int high) --
64
65 > void quicksort(treg reg[], int low, int high) --
66
67 > void regEliminados(treg reg[], int n) --
68
69 > void modReg(treg reg[], int n) --
70
71 > void MostrarArchivo() --
72
73 > void archivoBin(treg reg[], int n) --
74
75 > int cargarArchBin(treg reg[], int n) --
76
77 > void BinArchEl() --
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

6. ANEXOS

RGA_RP13_ANEXO



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo: implementación de algoritmos en C

:Pearson Educación. Sznajdleder, P. A. (2017)..

Buenos Aires, Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN: 9688804711

Programación en C. Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España: McGraw-Hill.

ISBN: 8448130138