



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 González

Matrícula: 372576

Maestro: Pedro Núñez Yépiz

Actividad No. : 12

Tema - Unidad : Archivos

Ensenada Baja California a 12 de noviembre del 2023

```

1 // Antonio Ramos Gonzalez ID: 372576
2 // 07/11/2023 || 12/11/2023
3 // Se generara un archivo donde se almacenara datos generados por el usuario, y la lectura de archivos
4 // RGA_Act12_932
5 #include "Babilonia.h"
6 #define N 2000
7 > typedef struct _Reg...
17 } Treg;
18
19 int msg();
20 void menu();
21 int verMt(Treg reg[], int n, int mt);
22 int cargar(Treg reg[], int n, char filename[]);
23 Treg agregar(Treg reg[], int i);
24 void nombreAl(char nombre[], int sex);
25 void apAl(char apellido[]);
26 int searchSec(Treg reg[], int n, int mt);
27 int searchBin(Treg reg[], int inf, int sup, int mt);
28 int orderBu(Treg reg[], int n);
29 void pintReg(Treg reg[], int n);
30 void pintOneReg(Treg reg[], int n);
31 void swap(Treg reg[], int i, int j);
32 void createTXT(Treg reg[], int n);
33 int partition(Treg reg[], int low, int high);
34 void quicksort(Treg reg[], int low, int high);
35 void RegEnArch();
36 void regEliminados(Treg reg[], int n);
37
38 int main()
39 {
40     srand(time(NULL));
41     fflush(stdin);
42     menu();
43     return 0;
44 }
45
46 int msg()
47 {
48     system("CLS");
49     printf("Menu\n");
50     printf("1.-Cargar\n");
51     printf("2.-Agregar\n");
52     printf("3.-Eliminar registro\n");
53     printf("4.-Buscar Registro\n");
54     printf("5.-Ordenar\n");
55     printf("6.-Imprimir\n");
56     printf("7.-Archivo texto\n");
57     printf("8.-Cantidad de registros\n");
58     printf("9.-Registros eliminados\n");
59     printf("0.-Salir\n");
60     return valid("Elije una opcion: ", 0, 9);
61 }
62
63 void menu()
64 {
65     int opc, kay;
66     int i = 0, j, band, bus;
67     int fileBand = 0;
68     Treg reg[N];

```

```

68 Treg reg[N];
69 do
70 {
71     opc = msg();
72     switch (opc)
73     {
74     case 1:
75         if (!fileBand)
76         {
77
78             if (i + 313 < N)
79             {
80                 i = cargar(reg, i, "datos.txt");
81                 fileBand = 1;
82                 band=1;
83             }
84             else
85             {
86                 printf("LA BASE DE DATOS HA SIDO LLENADA\n");
87                 system("PAUSE");
88             }
89         }
90         else
91         {
92             printf("YA SE HA CARGADO EL ARCHIVO\n");
93             system("PAUSE");
94         }
95         break;
96
97     case 2:
98         if (i < N)
99         {
100
101             for (j = 0; j < 10; j++)
102             {
103                 if (i < N)
104                 {
105                     reg[i] = agregar(reg, i);
106                     i++;
107                     band = 1;
108                 }
109             }
110         }
111         else
112         {
113             printf("Base de datos llena\n");
114             system("PAUSE");
115         }
116
117         break;
118     case 3:
119         kay = valid("Ingrese matricula que desea eliminar: ", 300000, 399999);
120         if (band)
121         {
122             bus = searchSec(reg, i, kay);
123             pintOneReg(reg, bus);
124         }
125         else
126         {

```

```

    {
        bus = searchBin(reg, 0, i, kay);
        pintOneReg(reg, bus);
    }
    if (valid("DESEA ELIMINAR ALUMNO\n1.-Si\n0.-NO\n", 0, 1))
    {
        reg[bus].status = 0;
    }

    break;
case 4:
    kay = valid("Ingrese matricula que desea Buscar: ", 300000, 399999);
    if (band)
    {
        bus = searchSec(reg, i, kay);
        pintOneReg(reg, bus);
    }
    else
    {
        bus = searchBin(reg, 0, i, kay);
        pintOneReg(reg, bus);
    }
    system("PAUSE");
    break;
case 5:
    if (band)
    {
        if (i < 500)
        {
            band = orderBu(reg, i);
        }
        else
        {
            band = 1;
            quicksort(reg, 0, i - 1);
        }
    }
    break;
case 6:
    pintReg(reg, i);
    break;
case 7:
    createTXT(reg, i);
    break;
case 8:
    RegEnArch();
    system("PAUSE");
    break;
case 9:
    regEliminados(reg, i);
    break;

```

```

} while (opc != 0);

```

```

}

```

```

void nombreAl(char nombre[], int sex)
{
    int i, al;
    char nombH[50][20] = {
        "JUAN", "PEDRO", "LUIS", "CARLOS", "ANDRES", "MIGUEL", "DAVID", "RAFAEL", "FRANCISCO",
        "ALEJANDRO", "JORGE", "DANIEL", "GUILLERMO", "SERGIO", "PABLO", "JAVIER", "ALBERTO",
        "SERGIO", "OSCAR", "GUILLERMO", "MARIO", "RICARDO", "IGNACIO", "FEDERICO", "RODRIGO",
        "MANUEL", "ANTONIO", "RAUL", "ALFREDO", "FERNANDO", "ROBERTO", "EDUARDO", "HUGO", "ENRIQUE",
        "EMILIO", "ARMANDO", "JULIO", "ADRIAN", "GUSTAVO", "ARTURO", "ANGEL", "HORACIO",
        "RODOLFO", "LUCAS", "RAMON", "OMAR", "CESAR", "SANTIAGO", "GERARDO", "ERNESTO"};
    char nombM[50][20] = {
        "ANA", "SOFIA", "CARMEN", "ISABEL", "ANDREA", "PATRICIA", "ELENA", "MARTA", "BEATRIZ",
        "PAULA", "JULIA", "VALERIA", "RAQUEL", "NATALIA", "CLAUDIA", "VALENTINA", "LORENA",
        "ELENA", "MARIANA", "ADRIANA", "LAURA", "ANTONIA", "MARTINA", "ANGELA", "LUISA",
        "SARA", "VERONICA", "ALICIA", "ROSA", "LUCIA", "VICTORIA", "GLORIA", "INES",
        "ROCIO", "VANESA", "RAFAELA", "ANAIS", "EVA", "SANDRA", "LOLA", "ANA", "ESTHER",
        "MIRIAM", "LIDIA", "YOLANDA", "INES", "MARINA", "CAROLINA", "PILAR", "IRENE"};
    for (i = 0; i < 10; i++)
    {
        al = rand() % 50;
    }
    if (sex == 1)
    {
        strcpy(nombre, nombH[al]);
    }
    else
    {
        strcpy(nombre, nombM[al]);
    }
}

```

```

void apAl(char apellido[])
{
    int i, al;
    char ap[87][20] =
    {
        "PEREZ", "LOPEZ", "GONZALEZ", "RODRIGUEZ", "MARTINEZ", "SANCHEZ", "FERNANDEZ", "GARCIA", "RODRIGUEZ", "DIAZ", "TORRES", "RAMIREZ", "JIMENEZ",
        "VARGAS", "MORALES", "RUIZ", "CASTRO", "HERRERA", "MEDINA", "NAVARRO", "DELGADO", "ROMERO", "MORENO", "GOMEZ", "VARGAS", "PEREZ", "FLORES", "GONZALEZ",
        "NUÑEZ", "SILVA", "SOTO", "MENDOZA", "ROJAS", "MENDEZ", "SALAS", "AGUIRRE", "BENITEZ", "GUZMAN", "PAREDES", "RIOS", "VALENZUELA", "CORDERO", "ESCOBAR",
        "OCHOA", "MORALES", "PIZARRO", "ARAYA", "CABRERA", "LOPEZ", "ORTEGA", "FERNANDEZ", "NAVARRO", "GUZMAN", "MENDEZ", "ROJAS", "SALAZAR", "VILLANUEVA", "PAREDES",
        "RIVAS", "SANDOVAL", "ESPINOZA", "MUÑOZ", "ACOSTA", "DELGADO", "CASTRO", "NAVARRO", "GONZALEZ", "MENDOZA", "SILVA", "RIOS", "ROJAS", "AGUIRRE", "MORALES", "GOMEZ",
        "SOTO", "TORRES", "PEREZ", "MEDINA", "BENITEZ", "GARCIA", "VARGAS", "RODRIGUEZ", "RAMIREZ", "TORRES", "MORALES", "HERRERA", "JIMENEZ"};
    for (i = 0; i < 10; i++)
    {
        al = rand() % 87;
    }
    strcpy(apellido, ap[al]);
}

```

```

Treg agregar(Treg reg[], int i)
{
    int sex, min = 18, max = 35;
    do
    {
        reg[i].kay = rand() % 100000 + 300000;
    } while (verMt(reg, i, reg[i].kay));

    sex = rand() % 2 + 1;
    if (sex == 1)
    {

```

```

Treg agregar(Treg reg[], int i)
{
    int sex, min = 18, max = 35;
    do
    {
        reg[i].kay = rand() % 100000 + 300000;
    } while (verMt(reg, i, reg[i].kay));

    sex = rand() % 2 + 1;
    if (sex == 1)
    {
        strcpy(reg[i].sex, "Masculino");
    }
    if (sex == 2)
    {
        strcpy(reg[i].sex, "Femenino");
    }

    nombreAl(reg[i].nombre, sex);

    apAl(reg[i].apP);
    apAl(reg[i].apM);

    reg[i].edad = rand() % (max - min + 1) + min;

    return reg[i];
}

int searchSec(Treg reg[], int n, int mt)
{
    int i;
    i = 0;
    for (i = 0; i <= n; i++)
    {
        if (reg[i].kay == mt)
        {
            return i;
        }
    }
    return -1;
}

int searchBin(Treg reg[], int inf, int sup, int mt)
{
    int med; // variable local
    while (inf <= sup)
    {
        med = (inf + sup) / 2;
        if (reg[med].kay == mt)
        {
            return med;
        }
        else
        {
            if (mt < reg[med].kay)
            {
                sup = med--;
            }
        }
    }
}

```

```

287     }
288     else
289     {
290         inf = med++;
291     }
292 }
293 }
294 return -1;
295 }
296
297 int orderBu(Treg reg[], int n)
298 {
299     int i, j;
300     Treg temp;
301
302     for (i = 0; i < n - 1; i++)
303     {
304         for (j = i + 1; j < n; j++)
305         {
306             if (reg[j].kay < reg[i].kay)
307             {
308                 temp = reg[i];
309                 reg[i] = reg[j];
310                 reg[j] = temp;
311             }
312         }
313     }
314     return 0;
315 }
316
317 void pintReg(Treg reg[], int n)
318 {
319     int i = 0, j = 0, elec;
320     system("PAUSE");
321     printf("%-6s %-15s %-23s %-15s %-10s %-10s %-10s\n",
322         "No.", "Matricula", "Nombre", "ApP", "ApM", "Edad", "Sexo");
323     do
324     {
325         i = 0;
326         if (j < n)
327         {
328             while (i < 40 && j < n)
329             {
330
331                 if (reg[j].status)
332                 {
333                     printf("%-6d.- %-10d %-10s %-15s %-13s %-10d %-5s\n",
334                         j + 1, reg[j].kay, reg[j].nombre, reg[j].apP,
335                         reg[j].apM, reg[i].edad, reg[j].sex);
336                 }
337                 i++;
338                 j++;
339             }
340             elec = valid("1.-SI\n0.-NO\nSEGUIR IMPRIMIENDO: ", 0, 1);
341         }
342         else
343         {
344             printf("LIMITE ALCANZADO\n");
345             elec = 0;
346         }

```

```
elec = 0;
```

```
} while (elec);  
system("PAUSE");  
}
```

```
void pintOneReg(Treg reg[], int n)
```

```
{  
    if (n != -1)  
    {  
        printf("MATRICULA: %d\nNOMBRE: %s \nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nEDAD: %d\nSEXO:%s\n",  
            reg[n].kay, reg[n].nombre, reg[n].apP,  
            reg[n].apM, reg[n].edad, reg[n].sex);  
    }  
    else  
    {  
        printf("Registro no encontrado\n");  
    }  
}
```

```
int verMt(Treg reg[], int n, int mt)
```

```
{  
    int i, j;  
    for (i = 0; i < n; i++)  
    {  
        for (j = 0; j <= i; j++)  
        {  
            if (reg[j].kay == mt)  
            {  
                return 1;  
            }  
        }  
    }  
    return 0;  
}
```

```
void createTXT(Treg reg[], int n)
```

```
{  
    FILE *fa;  
    int i = 0;  
    char namefile[30];  
  
    validCad("Ingrese nombre del arvhivo: ", namefile);  
    strcat(namefile, ".txt");  
    fa = fopen(namefile, "w");  
  
    for (i = 0; i < n; i++)  
    {  
        if (reg[i].status != 0)  
        {  
            fprintf(fa, "%d.- %-10d %-10s %-15s %-10s %-8d %s\n",  
                i + 1, reg[i].kay, reg[i].nombre, reg[i].apP,  
                reg[i].apM, reg[i].edad, reg[i].sex);  
        }  
    }  
    fclose(fa);  
}
```



```

void swap(Treg reg[], int i, int j)
{
    Treg temp = reg[i];
    reg[i] = reg[j];
    reg[j] = temp;
}

int partition(Treg reg[], int low, int high)
{
    Treg pivot;
    pivot.kay = reg[high].kay;
    int i = low - 1;

    for (int j = low; j <= high - 1; j++)
    {
        if (reg[j].kay <= pivot.kay)
        {
            i++;
            swap(reg, i, j);
        }
    }
    swap(reg, i + 1, high);
    return i + 1;
}

void quicksort(Treg reg[], int low, int high)
{
    if (low < high)
    {
        int pi = partition(reg, low, high);

        quicksort(reg, low, pi - 1);
        quicksort(reg, pi + 1, high);
    }
}

int cargar(Treg reg[], int n, char filename[])
{
    FILE *fa;
    Treg regTemp;
    int No = 0;
    fa = fopen(filename, "r");
    if (fa)
    {
        do
        {
            fscanf(fa, "%d.- %d %s %s %s %d %s", &No, &regTemp.kay, regTemp.nombre, regTemp.apP, regTemp.apM,
                &regTemp.edad, regTemp.sex);
            reg[n++] = regTemp;
        } while (!feof(fa));
        fclose(fa);
    }
    else
    {
        printf("ARCHIVO NO ENCONTRADO\n");
        system("PAUSE");
    }
}

```

```
    return n - 1;
```

```
}
```

```
void RegEnArch()
```

```
{
```

```
    int count;
```

```
    char fileName[50];
```

```
    char cmd[50];
```

```
    validCad("Ingrese el nombre del archivo ", fileName);
```

```
    system("mingw32-gcc-6.3.0.exe countTXT.c -o countTXT");
```

```
    sprintf(cmd, "countTXT.exe %s", fileName);
```

```
    count = system(cmd);
```

```
    if (count != -1)
```

```
    {
        printf("El archivo %s contiene %d registros\n", fileName, count);
    }
```

```
    else
```

```
    {
        printf("El archivo no fue encontrado\n");
    }
```

```
}
```

```
void regEliminados(Treg reg[], int n)
```

```
{
```

```
    FILE *fa;
```

```
    int i = 0;
```

```
    char namefile[30];
```

```
    validCad("Ingrese nombre del arvhivo: ", namefile);
```

```
    strcat(namefile, ".txt");
```

```
    fa = fopen(namefile, "w");
```

```
    fprintf(fa, "%-6s %-15s %-23s %-15s %-10s %-10s\n",
```

```
        "No.", "Matricula", "Nombre", "ApP", "ApM", "Edad", "Sexo");
```

```
    for (i = 0; i < n; i++)
```

```
    {
        if (reg[i].status == 0)
```

```
        {
            fprintf(fa, "%-6d %-10d %-10s %-15s %-10s %-8d %s\n",
                i + 1, reg[i].kay, reg[i].nombre, reg[i].apP,
                reg[i].apM, reg[i].edad, reg[i].sex);
        }
```

```
    }
```

```
    fclose(fa);
```

```
}
```

Menu

1.-Cargar

2.-Agregar

3.-Eliminar registro

4.-Buscar Registro

5.-Ordenar

6.-Imprimir

7.-Archivo texto

8.-Cantidad de registros

9.-Registros eliminados

0.-Salir

Elije una opcion: 1



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Menu

- 1.-Cargar
- 2.-Agregar
- 3.-Eliminar registro
- 4.-Buscar Registro
- 5.-Ordenar
- 6.-Imprimir
- 7.-Archivo texto
- 8.-Cantidad de registros
- 9.-Registros eliminados
- 0.-Salir

Elije una opcion: 4



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1	0.-	310442	MARIA	SOTO	PEREZ	33	FEMENINO
---	-----	--------	-------	------	-------	----	----------

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS

MATRICULA: 310442

NOMBRE: MARIA

APELLIDO PATERNO: SOTO

APELLIDO MATERNO: PEREZ

EDAD: 33

SEXO:FEMENINO

Presione una tecla para continuar . . .

No.	Matricula	Nombre		ApP	ApM	Edad	Sexo
1	.- 300109	ANA	SOTO	FLORES	18	Femenino	
2	.- 300383	JOFREY	GARCIA	SOTO	44	MASCULINO	
3	.- 300425	PEDRO	LOPEZ	CONTRERAS	26	MASCULINO	
4	.- 300429	KASSANDRA	ACOSTA	TORRES	20	FEMENINO	
5	.- 300518	ROSA	LOPEZ	YEPIZ	54	FEMENINO	
6	.- 300587	CRISTINA	TORRES	MORALES	43	FEMENINO	
7	.- 300620	ROSA	TORRES	ACOSTA	37	FEMENINO	
8	.- 300676	REYNA	NUNEZ	LOPEZ	18	FEMENINO	
9	.- 300785	LUPITA	TORRES	LOPEZ	21	FEMENINO	
10	.- 300809	VEGUETA	TORRES	GARCIA	43	MASCULINO	
11	.- 300811	CRISTINA	ACOSTA	ACOSTA	36	FEMENINO	
12	.- 300834	GOHAN	CONTRERAS	LOPEZ	28	MASCULINO	
13	.- 300856	KASSANDRA	ACOSTA	TORRES	52	FEMENINO	
14	.- 300858	PANFILA	NUNEZ	NUNEZ	53	FEMENINO	
15	.- 300941	JOFREY	MORALES	ACOSTA	50	MASCULINO	
16	.- 301131	PEDRO	CONTRERAS	GARCIA	24	MASCULINO	
17	.- 301145	MARIA	PEREZ	NUNEZ	33	FEMENINO	
18	.- 301164	KASSANDRA	MORALES	LOPEZ	33	FEMENINO	
19	.- 301185	BRENDA	PEREZ	YEPIZ	18	FEMENINO	
20	.- 301258	KASSANDRA	LOPEZ	TORRES	48	FEMENINO	
21	.- 301423	KASSANDRA	TORRES	MORALES	37	FEMENINO	
22	.- 301430	PEDRO	TORRES	CONTRERAS	30	MASCULINO	
23	.- 301469	LUNA	NUNEZ	TORRES	33	FEMENINO	
24	.- 301877	ROSA	ACOSTA	CONTRERAS	49	FEMENINO	
25	.- 302053	GOKU	TORRES	ACOSTA	28	MASCULINO	
26	.- 302187	GOTEM	NUNEZ	ACOSTA	38	MASCULINO	
27	.- 302269	ROSA	TORRES	MORALES	28	FEMENINO	
28	.- 302376	GOHAN	SOTO	LOPEZ	25	MASCULINO	
29	.- 302505	ROSA	MORALES	CONTRERAS	15	FEMENINO	
30	.- 302522	KAKAROTO	SOTO	CONTRERAS	40	MASCULINO	
31	.- 302532	GOKU	LOPEZ	LOPEZ	15	MASCULINO	
32	.- 302724	PANFILA	LOPEZ	GARCIA	47	FEMENINO	
33	.- 302898	REYNA	YEPIZ	PEREZ	43	FEMENINO	
34	.- 303024	BRENDA	YEPIZ	SOTO	16	FEMENINO	
35	.- 303036	LUNA	YEPIZ	SOTO	33	FEMENINO	
36	.- 303157	KAKAROTO	CONTRERAS	GARCIA	29	MASCULINO	
37	.- 303171	JONH	SOTO	CONTRERAS	18	MASCULINO	
38	.- 303176	EDRAD	MORALES	TORRES	19	MASCULINO	
39	.- 303448	PEDRO	PEREZ	SOTO	34	MASCULINO	
40	.- 303450	VEGUETA	LOPEZ	NUNEZ	18	MASCULINO	

1.-SI

0.-NO

SEGUIR IMPRIMIENDO: ☐

No.	Matricula	Nombre		ApP	ApM	Edad	Sexo
41	.- 303532	EDRAD	CONTRERAS	CONTRERAS	18	MASCULINO	
42	.- 303559	GOKU	PEREZ	NUNEZ	44	MASCULINO	
43	.- 303561	LUPITA	CONTRERAS	LOPEZ	26	FEMENINO	
44	.- 303704	MASIOSARE	NUNEZ	ACOSTA	20	MASCULINO	
45	.- 303792	ALICIA	CONTRERAS	YEPIZ	54	FEMENINO	
46	.- 303880	KAKAROTO	GARCIA	TORRES	43	MASCULINO	
47	.- 303985	VEGUETA	SOTO	NUNEZ	37	MASCULINO	
48	.- 304133	GOTEM	YEPIZ	YEPIZ	18	MASCULINO	
49	.- 304292	LUPITA	GARCIA	NUNEZ	21	FEMENINO	
50	.- 304385	VEGUETA	SOTO	GARCIA	43	MASCULINO	
51	.- 304504	ALICIA	ACOSTA	PEREZ	36	FEMENINO	
52	.- 304532	MARIA	TORRES	GARCIA	28	FEMENINO	
53	.- 304642	REYNA	SOTO	SOTO	52	FEMENINO	
54	.- 304721	MARIA	MORALES	YEPIZ	53	FEMENINO	
55	.- 304740	JONH	PEREZ	YEPIZ	50	MASCULINO	
56	.- 305102	MASIOSARE	ACOSTA	CONTRERAS	24	MASCULINO	
57	.- 305270	CESAR	RODRIGUEZ	MENDOZA	33	Masculino	
58	.- 305320	GOTEM	YEPIZ	NUNEZ	33	MASCULINO	
59	.- 305495	LUNA	ACOSTA	PEREZ	18	FEMENINO	
60	.- 305507	ALICIA	YEPIZ	LOPEZ	48	FEMENINO	
61	.- 305550	MASIOSARE	PEREZ	TORRES	37	MASCULINO	
62	.- 305560	KASSANDRA	NUNEZ	NUNEZ	30	FEMENINO	
63	.- 305628	EDRAD	ACOSTA	ACOSTA	33	MASCULINO	
64	.- 305769	ROSA	LOPEZ	CONTRERAS	49	FEMENINO	
65	.- 305809	LUNA	TORRES	LOPEZ	28	FEMENINO	
66	.- 305964	MARIA	MORALES	MORALES	38	FEMENINO	
67	.- 306087	KASSANDRA	GARCIA	PEREZ	28	FEMENINO	
68	.- 306144	GOHAN	SOTO	CONTRERAS	25	MASCULINO	
69	.- 306242	MARIA	CONTRERAS	CONTRERAS	15	FEMENINO	
70	.- 306294	EDRAD	CONTRERAS	YEPIZ	40	MASCULINO	
71	.- 306359	MASIOSARE	ACOSTA	CONTRERAS	15	MASCULINO	
72	.- 306471	KAKAROTO	LOPEZ	NUNEZ	47	MASCULINO	
73	.- 306513	CRISTINA	MORALES	YEPIZ	43	FEMENINO	
74	.- 307068	GOTEM	GARCIA	MORALES	16	MASCULINO	
75	.- 307152	BRENDA	TORRES	PEREZ	33	FEMENINO	
76	.- 307203	VEGUETA	MORALES	LOPEZ	29	MASCULINO	
77	.- 307267	MASIOSARE	NUNEZ	NUNEZ	18	MASCULINO	
78	.- 307674	GOTEM	SOTO	MORALES	19	MASCULINO	
79	.- 307775	ROSA	SOTO	CONTRERAS	34	FEMENINO	
80	.- 308035	LUPITA	ACOSTA	ACOSTA	18	FEMENINO	



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Menu

- 1.-Cargar
 - 2.-Agregar
 - 3.-Eliminar registro
 - 4.-Buscar Registro
 - 5.-Ordenar
 - 6.-Imprimir
 - 7.-Archivo texto
 - 8.-Cantidad de registros
 - 9.-Registros eliminados
 - 0.-Salir
- Elije una opcion: 8



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingrese el nombre del archivo hola

El archivo HOLA contiene 323 registros

Presione una tecla para continuar . . .

Menu

- 1.-Cargar
- 2.-Agregar
- 3.-Eliminar registro
- 4.-Buscar Registro
- 5.-Ordenar
- 6.-Imprimir
- 7.-Archivo texto
- 8.-Cantidad de registros
- 9.-Registros eliminados
- 0.-Salir

Elije una opcion: 3

MATRICULA: 310442

NOMBRE: MARIA

APELLIDO PATERNO: SOTO

APELLIDO MATERNO: PEREZ

EDAD: 33

SEXO: FEMENINO

DESEA ELIMINAR ALUMNO

1.-Si

0.-NO





Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

PROBLEMS

1

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese el nombre del archivo regeliminados
El archivo REGELIMINADOS contiene 1 registros
Presione una tecla para continuar . . .



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

countTXT.c

// Antonio Ramos Gonzalez Mt: 372576

// 07/11/2023 || 12/11/2023

// Se generara un archivo donde se almacenara datos generados por el usuario, y la lectura de archivos

// RGA_Act12_932

#include <stdio.h>

#include <string.h>

int main(int args, char *arg[])

{

FILE *fa;

int i = 0;

char fileName[50];

int tempNum;

int tempNum2;

char nomb[20], apP[20], apM[20];

int edad;

char sex[10];

strcpy(fileName, arg[1]);

strcat(fileName, ".txt");

fa = fopen(fileName, "r");

if (fa)

{

do

{

fscanf(fa, "%d.- %d %s %s %s %d %s",

&tempNum, &tempNum2, nomb, apP, apM, &edad, sex);

i++;

} while (!feof(fa));

fclose(fa);

return i-1;

}

else

{

return -1;

}

return 0;

}



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño