



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño



**Nombre del alumno:**

Luis Fernando Ochoa Angulo

**Matricula:**

372746

**Semestre:**

3ro- Computación

**Grupo:**

432

**Materia:**

Programación Estructurada

**Actividad:**

Actividad 9

**Nombre del Profesor:**

Pedro Nuñez Yepiz

**Fecha:** Martes 17 de Octubre del 2023



```
1 //Ochoa Angulo Luis Fernando 372746
2 //11-10-23 17-10-23
3 //Algoritmo con un menu para registros
4 //LFOA_Act10_01_432
5
6 #include <stdio.h>
7 #include <string.h>
8 #include <stdlib.h>
9 #include <time.h>
10 #include "8a.h"
11
12 #define N 500
13
14 typedef struct _alum
15 {
16     int status;
17     int matri;
18     char nombre[30];
19     char appater[30];
20     char apmater[30];
21     int sexo;
22     int edad;
23 } Talum;
24
25 Talum gen_reg_man();
26 Talum gen_rega_alea();
27 Talum eliminar_regs(Talum vect[], int *n);
28 void imp_registros(Talum vect[], int n);
29 void ordenar_reg(Talum vect[], int n);
30 void buscar_reg(Talum vect[], int n);
31 void menu();
32
33 int main()
34 {
35     menu();
36     return 0;
37 }
38 void menu()
39 {
40     int x = 0;
41     int i = 0, opc;
42     Talum vect[500];
43     do
44     {
45         printf("----- M E N U ----- \n");
46         printf("1.- AGREGAR (AUTOM 10 REGISTROS)\n");
47         printf("2.- AGREGAR MANUAL\n");
48         printf("3.- ELIMINAR REGISTRO (lógico)\n");
49         printf("4.- BUSCAR\n");
50         printf("5.- ORDENAR\n");
51         printf("6.- IMPRIMIR\n");
52         printf("0.- SALIR\n ");
```



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

```
53     opc = valid_num(0, 6, "Elije la opcion: ");
54     int j = i + 10;
55     switch (opc)
56     {
57     case 0:
58         opc = 0;
59     case 1:
60         for (i; i < j; i++)
61         {
62             vect[i] = gen_rega_alea();
63         }
64         printf("%d", i);
65         break;
66     case 3:
67         eliminar_regs(vect, &i);
68         break;
69     case 2:
70         vect[i] = gen_reg_man();
71         i++;
72         break;
73     case 4:
74         buscar_reg(vect,i);
75         break;
76     case 5:
77         ordenar_reg(vect, i);
78         break;
79     case 6:
80         imp_registros(vect, i);
81         break;
82     default:
83         break;
84     }
85
86     } while (opc != 0);
87 }
88
89 Talum gen_reg_man()
90 {
91     Talum regis;
92     regis.status = 1;
93     regis.matri = valid_num(300000, 399999, "Ingresa tu martirucla");
94     printf("Ingresa tu nombre: ");
95     fflush(stdin);
96     gets(regis.nombre);
97     printf("Ingresa tu apellido parterno: ");
98     fflush(stdin);
99     gets(regis.appater);
100    printf("Ingresa tu apellido materno: ");
101    fflush(stdin);
```

```
103     printf("Presiona: \n");
104     printf("1 - Hombre \n");
105     printf("2 - Mujer \n");
106     regis.sexo = valid_num(1, 2, "Ingresa tu sexo");
107     regis.edad = valid_num(1, 100, "Ingresa tu edad");
108     return regis;
109 }
```

```
1  Talum gen_rega_alea()
2  {
3      Talum regis;
4
5      char *nHombre[10] = {"Yepiz", "Enrique", "Pedro", "Tonysinz", "Daniel", "Miguel", "Alejandro", "José", "Javier", "Antonio"};
6      char *nMujer[10] = {"Elizabeth", "Valentina", "Jazmin", "Sofia", "Yaquelin", "Maria", "Dana", "Raquel", "Denisse", "Karen"};
7      char *apellidos[20] = {"Flores", "Morales", "Vazquez", "Jimenez", "Reyes", "Diaz", "Torres", "Gutierrez", "Ruiz", "Mendoza",
8                             "Gonzalez", "Lopez", "Rodriguez", "Martinez", "Valencia", "Fernandez", "Gomez", "Sanchez", "Ramirez", "Torres"};
9
10     regis.status = 1;
11     regis.matri = (rand() % 100000) + 300000;
12     int sexo = rand() % 100;
13     if (sexo%2 == 1)
14     {
15         strcpy(regis.nombre, nHombre[rand() % 10]);
16         regis.sexo = 1;
17     }
18     else
19     {
20         strcpy(regis.nombre, nMujer[rand() % 10]);
21         regis.sexo = 2;
22     }
23     strcpy(regis.appater, apellidos[rand() % 20]);
24     strcpy(regis.apmater, apellidos[rand() % 20]);
25     regis.edad = (rand() % 88) + 13;
26     return regis;
27 }
28 Talum eliminar_reg(Talum vect[], int *n)
29 {
30     int matri = valid_num(300000, 399999, "Ingresa matricula del registro que quieres borrar: ");
31     int k = 0;
32     Talum vect[N];
33
34     for (int i = 0; i < *n; i++)
35     {
36         if (matri != vect[i].matri)
37         {
38             vect[k] = vect[i];
39             k++;
40         }
41     }
42
43     if (k == *n)
44     {
45         printf("La matricula no esta en el registro \n");
46         return vect[N];
47     }
48     else
49     {
50         printf("Registro con matricula %d borrado.\n", matri);
51         *n = k;
52         for (int i = 0; i < k; i++)
53         {
54             vect[i] = vect[i];
55         }
56         return vect[k];
57     }
58 }
```



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

```
void ordenar_reg(Talum vect[], int n)
{
    int i, j;
    Talum temp;
    for (i = 0; i < n - 1; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (vect[j].matri < vect[i].matri)
            {
                temp = vect[i];
                vect[i] = vect[j];
                vect[j] = temp;
            }
        }
    }
}

void buscar_reg(Talum vect[], int n)
{
    int i;
    int matri = valid_num(300000, 399999, "Ingresa la matricula del registro que deseas buscar: ");

    for (i = 0; i < n; i++)
    {
        if (matri == vect[i].matri)
        {
            printf("Se encontro el usuario con matricula %d en el registro.\n", matri);
            printf("Estos son sus datos personales: ");

            printf("-----\n");
            printf("|  n°  |  matricula  |  nombre  |  ap.paterno  |  ap.materno  |  sexo  |  edad  |\n");

            printf("-----\n");
            printf("|  %4d  |  %11d  |  %12s  |  %12s  |  %13s  |  %5d  |  %5d  |\n", i, vect[i].matri,
            vect[i].nombre, vect[i].appater, vect[i].apmater, vect[i].sexo, vect[i].edad);
            printf("-----\n");

            return;
        }
    }

    if(i == n)
    {
        printf("No existe el usuario con matricula %d", matri);
    }
}
```