



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. 10: ACTIVIDAD 10

Tema - Unidad 7: Estructuras

Ensenada Baja California a 15 de octubre del 2023

```

1 // Antonio Ramos Gonzalez Mt: 372576
2 // 10/9/2023 || 15/13/2023
3 // En esta Practica se definira un tipo de variable como un struct, para ingresar datos 315985 de distintos alumnos con un menu
4 // RGA_Act10_932
5 #include "Babilonia.h"
6 #define N 500
7 typedef struct _alumn
8 {
9     char name[30];
10    char lasP[30];
11    char lasM[30];
12    int mt;
13    int age;
14    char sex[10];
15    int status;
16 } Talum;
17
18 int msg();
19 void nombAl(char nombre[], int sex);
20 void apellidoAl(char apellido[]);
21 void menu();
22 void printReg(Talum alumn[], int n);
23 int order_reg(Talum alumn[], int n, int band);
24 int searchSec(Talum alumn[], int n, int mt);
25 int searchBin(Talum alumn[], int inf, int sup, int mt);
26 Talum genAl(Talum alumn[], int i);
27 Talum llenarManual(Talum alumn[], int i);
28 Talum eliminar(Talum alumn);
29 void busq(int ind);
30 int verMt(Talum alumn[], int n, int mt);
31
32 int main()
33 {
34     srand(time(NULL));
35     menu();
36     return 0;
37 }
38
39 // Muestra mensaje
40 int msg()
41 {
42     system("CLS");
43     printf("MENU\n");
44     printf("1.-Agrega alumnar(AUTOM 10 alumnos)\n");
45     printf("2.-Agrega alumno manual\n");
46     printf("3.-Eliminar alumno(logico)\n");
47     printf("4.-Buscar\n");
48     printf("5.-Ordenar\n");
49     printf("6.-Imprimir\n");
50     printf("0.-Salir\n");
51     return valid("Elije una opcion: ", 0, 6);
52 }
53
54 // Menu de elecciones
55 void menu()
56 {
57     Talum reg[N];
58     int opc, el, i = 0, j;
59     int mt, bus;

```

```

54 // Meno de elecciones
55 void menu()
56 {
57     Talum reg[N];
58     int opc, el, i = 0, j;
59     int mt, bus;
60     bool band;
61
62     do
63     {
64         opc = msg();
65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)
73                         {
74                             reg[i] = genAl(reg, i);
75                             i++;
76                         }
77                     }
78                     band = 1;
79                 }
80                 else
81                 {
82                     printf("BASE DE DATOS LLENA\n");
83                     system("PAUSE");
84                 }
85                 break;
86
87             case 2:
88                 if (i < N)
89                 {
90                     reg[i] = llenarManual(reg, i);
91                     i++;
92                     band = 1;
93                 }
94                 else
95                 {
96                     printf("BASE DE DATOS LLENA\n");
97                     system("PAUSE");
98                 }
99                 break;
100
101             case 3:
102                 el = valid("Ingrese matricula: ", 300000, 399999);
103                 if (band == 1)
104                 {
105                     bus = searchSec(reg, i, el);
106                 }
107                 else
108                 {
109                     bus = searchBin(reg, 0, i, el);
110                 }
111                 reg[bus].status=0;
112                 break;

```

```

113
114     case 4:
115         mt = valid("Ingresa Matricula: ", 300000, 399999);
116
117         if (band == 1)
118         {
119             bus = searchSec(reg, i, mt);
120         }
121         else
122         {
123             bus = searchBin(reg, 0, i, mt);
124         }
125
126         busq(bus);
127
128         break;
129
130     case 5:
131
132         if (band == 1)
133         {
134             band = order_reg(reg, i, band);
135         }
136
137         break;
138
139     case 6:
140         printReg(reg, i);
141         break;
142     }
143     } while (opc != 0);
144 }
145

```

```

146 // Funcion para nombres
147 void nombAl(char nombre[], int sex)
148 {
149     int fil;
150     char nombM[10][10] =
151     {
152         "ANA",
153         "JULIETA",
154         "JOSEFINA",
155         "DANIELA",
156         "CARMEN",
157         "SOFIA",
158         "LAURA",
159         "ANDREA",
160         "ELENA",
161         "ISABEL"};
162
163     char nombH[10][10] =
164     {
165         "JUAN",
166         "CARLOS",
167         "ROBERTO",
168         "DAMIAN",
169         "ANDRES",
170         "DAVID",

```

```

        "ROBERTO",
        "DAMIAN",
        "ANDRES",
        "DAVID",
        "ALEJANDR",
        "MIGUEL",
        "PEDRO",
        "FERNANDO"};

    if (sex == 1)
    {
        fil = rand() % 10;
        strcpy(nombre, nombH[fil]);
    }
    if (sex == 2)
    {
        fil = rand() % 10;
        strcpy(nombre, nombM[fil]);
    }
}

```

```

// Funcio para apellidos
void apellidoAl(char apellido[])
{
    int fil;
    char ap[10][15] =
    {
        "GARCIA",
        "RODRIGUEZ",
        "PEREZ",
        "LOPEZ",
        "MARTINEZ",
        "GONZALEZ",
        "SANCHEZ",
        "ROMERO",
        "FRENANDEZ",
        "TORRES"};

    fil = rand() % 10;
    strcpy(apellido, ap[fil]);
}

```

```

// Funcion que verifica que no se repitan matriculas
int verMt(Talum alumn[], int n, int mt)
{
    int i, j, cont;
    cont = 0;
    for (i = 0; i < n; i++)
    {
        for (j = 0; j <= i; j++)
        {
            if (alumn[j].mt == mt)
            {
                cont = 1;
            }
        }
    }
    if (cont == 1)
    {
        return 0;
    }
}

```

```

    {
        return 1;
    }
else
{
    return 0;
}
}

```

```

// Funcion que genera datos de alumnos aleatorios
Talum genAl(Talum alumn[], int i)

```

```

{
    int sex, val;
    char nombre[10], apellido[15];

    sex = rand() % 2 + 1;

    nombAl(nombre, sex);
    strcpy(alumn[i].name, nombre);

    apellidoAl(apellido);
    strcpy(alumn[i].lasP, apellido);

    apellidoAl(apellido);
    strcpy(alumn[i].lasM, apellido);

    alumn[i].age = rand() % 13 + 18;

    alumn[i].status = 1;

    do
    {
        val = 0;
        alumn[i].mt = rand() % 1000000 + 300000;
        val = verMt(alumn, i, alumn[i].mt);
    } while (val != 0);

    if (sex == 1)
    {
        strcpy(alumn[i].sex, "H");
    }
else
{
    strcpy(alumn[i].sex, "M");
}

    return alumn[i];
}

```

```

// Funcion para ingresar a un alumno de manera manual
Talum llenarManual(Talum alumn[], int i)

```

```

{
    int sex, val;
    system("CLS");
    fflush(stdin);
    do
    {

```

```

system("CLS");
fflush(stdin);
do
{
    val = 0;
    alumn[i].mt = valid("Ingreso matricula: ", 300000, 399999);
    val = verMt(alumn, i, alumn[i].mt);

    if (val != 0)
    {
        printf("Matricula ya existente\n");
    }

} while (val != 0);

validCad("Ingreso nombre: ", alumn[i].name);
validCad("Ingreso apellido paterno: ", alumn[i].lasP);
validCad("Ingreso apellido materno: ", alumn[i].lasM);

system("CLS");
alumn[i].age = valid("Ingreso Edad: ", 18, 30);

sex = valid("Ingresa el sexo(1.-H,2.-M): ", 1, 2);

if (sex == 1)
{
    strcpy(alumn[i].sex, "H");
}
else
{
    strcpy(alumn[i].sex, "M");
}
alumn[i].status = 1;

return alumn[i];
}

// Funcion de busqueda secuencial
int searchSec(Talum alumn[], int n, int mt)
{
    int i;
    i = 0;
    for (i = 0; i <= n; i++)
    {
        if (alumn[i].mt == mt)
        {
            return i;
        }
    }
    return -1;
}

// Funcion que ordena el vector
int order_reg(Talum alumn[], int n, int band)
{
    int i, j;
    Talum temp;

    for (i = 0; i < n - 1; i++)

```

```

for (i = 0; i < n - 1; i++)
{
    for (j = i + 1; j < n; j++)
    {
        if (alumn[j].mt < alumn[i].mt)
        {
            temp = alumn[i];
            alumn[i] = alumn[j];
            alumn[j] = temp;
        }
    }
}
return band = 0;
}

```

```

// Funcion para imprimir el vector
void printReg(Talum alumn[], int n)
{
    int i;
    system("CLS");
    printf("%-10s %-10s %-10s %-10s %-4s %-5s\n",
        "Matricula", "Nombre", "ApP", "ApM", "Edad", "Sexo");
    for (i = 0; i < n; i++)
    {
        if (alumn[i].status == 1)
        {
            printf("%-10d %-10s %-10s %-10s %-4d %-5s\n",
                alumn[i].mt, alumn[i].name, alumn[i].lasP,
                alumn[i].lasM, alumn[i].age, alumn[i].sex);
        }
    }
    system("PAUSE");
}

```

```

// Funcion que imprime el alumno buscado
void busq(int ind)
{
    if (ind != -1)
    {
        printf("Alumno encontrado en el indice: %d\n", ind+1);
    }
    else
    {
        printf("Alumno no encontrado\n");
    }
    system("PAUSE");
}

```

```

// Funcion de busqueda binaria
int searchBin(Talum alumn[], int inf, int sup, int mt)
{

```



```
// Funcion de busqueda binaria
```

```
int searchBin(Talum alumn[], int inf, int sup, int mt)
{
    int med; // variable local
    while (inf <= sup)
    {
        med = (inf + sup) / 2;
        if (alumn[med].mt == mt)
        {
            return med;
        }
        else
        {
            if (mt < alumn[med].mt)
            {
                sup = med--;
            }
            else
            {
                inf = med++;
            }
        }
    }
    return -1;
}
```

```
61
62     do
63     {
64 //      opc = msg();
65       switch (opc)
66       {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

- 1.-Agregar alumnar(AUTOM 10 alumnos)
- 2.-Agregar alumno manual
- 3.-Eliminar alumno(logico)
- 4.-Buscar
- 5.-Ordenar
- 6.-Imprimir
- 0.-Salir

Elije una opcion: 1

```
59     int mt, bus;  
60     bool band;  
61  
62     do  
63     {  
64         opc = msg();  
65         switch (opc)  
66         {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

- 1.-Agregar alumnar(AUTOM 10 alumnos)
- 2.-Agregar alumno manual
- 3.-Eliminar alumno(logico)
- 4.-Buscar
- 5.-Ordenar
- 6.-Imprimir
- 0.-Salir

Elije una opcion: 2

```
58     int opc, el, i = 0, j;  
59     int mt, bus;  
60     bool band;  
61  
62     do  
63     {  
64         opc = msg();  
65         switch (opc)  
66         {
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese matricula: 372576

```
64  opc = msg();
65  switch (opc)
66  {
67  case 1:
68      if (i < N)
69      {
70          for (j = 0; j < 10; j++)
71          {
72              if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese nombre: ANTONIO

```
62      do
63      {
64          opc = msg();
65          switch (opc)
66          {
67              case 1:
68                  if (i < N)
69                  {
70                      for (j = 0; j < 10; j++)
71                      {
72                          if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese apellido paterno: RAMOS

```
62 do
63 {
64     opc = msg();
65     switch (opc)
66     {
67     case 1:
68         if (i < N)
69         {
70             for (j = 0; j < 10; j++)
71             {
72                 if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

ingrese apellido materno: GONZALEZ

```
65      switch (opc)
66      {
67      case 1:
68      //      if (i < N)
69      {
70          for (j = 0; j < 10; j++)
71          {
72              if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese Edad: 19


```
62     do
63     {
64         opc = msg();
65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingresa el sexo(1.-H,2.-M): 1

```

65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

1.-Agrega alumnar(AUTOM 10 alumnos)

2.-Agrega alumno manual

3.-Eliminar alumno(logico)

4.-Buscar

5.-Ordenar

6.-Imprimir

0.-Salir

Elije una opcion: 6

```

65      switch (opc)
66      {
67      case 1:
68      //      if (i < N)
69      {
70          for (j = 0; j < 10; j++)
71          {
72              if (i < N)

```

| PROBLEMS | OUTPUT | DEBUG CONSOLE | <u>TERMINAL</u> | PORTS |
|---|----------|---------------|-----------------|-----------|
| Matricula | Nombre | ApP | ApM | Edad Sexo |
| 318420 | JOSEFINA | GARCIA | PEREZ | 19 M |
| 312004 | ISABEL | MARTINEZ | GARCIA | 29 M |
| 328866 | FERNANDO | TORRES | TORRES | 24 H |
| 309650 | FERNANDO | RODRIGUEZ | TORRES | 25 H |
| 328510 | MIGUEL | FRENANDEZ | FRENANDEZ | 22 H |
| 311942 | ELENA | ROMERO | RODRIGUEZ | 29 M |
| 317824 | CARLOS | FRENANDEZ | TORRES | 22 H |
| 331504 | DANIELA | ROMERO | TORRES | 19 M |
| 320646 | ELENA | MARTINEZ | GONZALEZ | 20 M |
| 312187 | JULIETA | MARTINEZ | GONZALEZ | 26 M |
| 372576 | ANTONIO | RAMOS | GONZALEZ | 19 H |
| Presione una tecla para continuar . . . | | | | |

```
66         {
67         case 1:
68         // if (i < N)
69         {
70         for (j = 0; j < 10; j++)
71         {
72         if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

- 1.-Agregar alumnar(AUTOM 10 alumnos)
- 2.-Agregar alumno manual
- 3.-Eliminar alumno(logico)
- 4.-Buscar
- 5.-Ordenar
- 6.-Imprimir
- 0.-Salir

Elije una opcion: 3

```
61
62     do
63     {
64         opc = msg();
65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingrese matricula: 372576

```

64     opc = msg();
65     switch (opc)
66     {
67     case 1:
68         if (i < N)
69         {
70             for (j = 0; j < 10; j++)
71             {
72                 if (i < N)

```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

| Matricula | Nombre | ApP | ApM | Edad | Sexo |
|-----------|----------|-----------|-----------|------|------|
| 318420 | JOSEFINA | GARCIA | PEREZ | 19 | M |
| 312004 | ISABEL | MARTINEZ | GARCIA | 29 | M |
| 328866 | FERNANDO | TORRES | TORRES | 24 | H |
| 309650 | FERNANDO | RODRIGUEZ | TORRES | 25 | H |
| 328510 | MIGUEL | FRENADEZ | FRENADEZ | 22 | H |
| 311942 | ELENA | ROMERO | RODRIGUEZ | 29 | M |
| 317824 | CARLOS | FRENADEZ | TORRES | 22 | H |
| 331504 | DANIELA | ROMERO | TORRES | 19 | M |
| 320646 | ELENA | MARTINEZ | GONZALEZ | 20 | M |
| 312187 | JULIETA | MARTINEZ | GONZALEZ | 26 | M |

Presione una tecla para continuar . . .

```
66      {
67      case 1:
68      if (i < N)
69      {
70      for (j = 0; j < 10; j++)
71      {
72      if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

1.-Agrega alumnar(AUTOM 10 alumnos)

2.-Agrega alumno manual

3.-Eliminar alumno(logico)

4.-Buscar

5.-Ordenar

6.-Imprimir

0.-Salir

Elije una opcion: 4

```
63      {
64      //      opc = msg();
65      switch (opc)
66      {
67      case 1:
68      //      if (i < N)
69      {
70          for (j = 0; j < 10; j++)
71          {
72              if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingresar Matricula: 317824


```
63      {
64          opc = msg();
65          switch (opc)
66          {
67              case 1:
68                  if (i < N)
69                  {
70                      for (j = 0; j < 10; j++)
71                      {
72                          if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Alumno encontrado en el indice: 7

Presione una tecla para continuar . . .

```
64      opc = msg();
65      switch (opc)
66      {
67      case 1:
68          if (i < N)
69          {
70              for (j = 0; j < 10; j++)
71              {
72                  if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

1.-Agrega alumnar(AUTOM 10 alumnos)

2.-Agrega alumno manual

3.-Eliminar alumno(logico)

4.-Buscar

5.-Ordenar

6.-Imprimir

0.-Salir

Elije una opcion: 5

```

67         case 1:
68             if (i < N)
69             {
70                 for (j = 0; j < 10; j++)
71                 {
72                     if (i < N)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

| Matricula | Nombre | ApP | ApM | Edad | Sexo |
|-----------|----------|-----------|-----------|------|------|
| 309650 | FERNANDO | RODRIGUEZ | TORRES | 25 | H |
| 311942 | ELENA | ROMERO | RODRIGUEZ | 29 | M |
| 312004 | ISABEL | MARTINEZ | GARCIA | 29 | M |
| 312187 | JULIETA | MARTINEZ | GONZALEZ | 26 | M |
| 317824 | CARLOS | FRENANDEZ | TORRES | 22 | H |
| 318420 | JOSEFINA | GARCIA | PEREZ | 19 | M |
| 320646 | ELENA | MARTINEZ | GONZALEZ | 20 | M |
| 328510 | MIGUEL | FRENANDEZ | FRENANDEZ | 22 | H |
| 328866 | FERNANDO | TORRES | TORRES | 24 | H |
| 331504 | DANIELA | ROMERO | TORRES | 19 | M |

Presione una tecla para continuar . . .

```
62     do
63     {
64         opc = msg();
65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

MENU

- 1.-Agrega alumnar(AUTOM 10 alumnos)
- 2.-Agrega alumno manual
- 3.-Eliminar alumno(logico)
- 4.-Buscar
- 5.-Ordenar
- 6.-Imprimir
- 0.-Salir

Elije una opcion: 4

```
62     do
63     {
64         opc = msg();
65         switch (opc)
66         {
67             case 1:
68                 if (i < N)
69                 {
70                     for (j = 0; j < 10; j++)
71                     {
72                         if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Ingresar Matricula: 317824

```
63 {
64     opc = msg();
65     switch (opc)
66     {
67     case 1:
68         if (i < N)
69         {
70             for (j = 0; j < 10; j++)
71             {
72                 if (i < N)
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

Alumno encontrado en el indice: 5

Presione una tecla para continuar . . .