



Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniero en Computación

Asignatura:

Programación Estructurada

Actividad 10:

**FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA
ESTRUCTURAS Y LIBRERIAS**

Brayan Arturo Rocha Meneses

Matricula:

371049

Ensenada Baja California 17 de Octubre del 2023

```

#include "2lives.h"

int msges();
void menu();

typedef struct _alumno{
    int status;
    int matricula;
    char nombre[100];
    int edad;
    int sexo;
} Talumno;

Talumno alumno[10];
Talumno GenerarManual(void);
Talumno GenerarAutom(void);
int AlumnosEliminados[500];
int CantidadEliminados = 0;
int rango(int ri, int rf);
int GenerarMatriculaRandom();
int GenerarEdadRandom();
char GenerarNombreRandom(char nombre[]);
int OrdenarMatricula(Talumno alumnos[], int cantidad);
int BuscarMatricula(Talumno alumnos[], int BuscarMatricula, int cantidad);
int BusquedaBinaria(Talumno alumnos[], int i, int d, int Matricula);

char PriNomFem[10][10] = {"JUAN", "ALEJANDRO", "MIGUEL", "FRANCISCO", "MANUEL", "ROBERTO", "ARTURO", "SERGIO", "TONY", "EYNAR"};
char SegNomFem[10][15] = {"CARLOS", "JOSE", "ANGEL", "JAVIER", "ALEJANDRO", "DANIEL", "JUAN", "GEOVANY", "GABRIEL", "ALBERTO"};
char PriNomMas[10][10] = {"LUIS", "PEDRO", "JOSE", "DAVID", "RAUL", "ROBERTO", "DIEGO", "CARLOS", "EDUARDO ANTONIO", "EDUARDO"};

char SegNomMas[10][10] = {"EDUARDO", "LUIS", "RAFAEL", "ENRIQUE", "ARTURO", "FRANCISCO", "SAID", "MANUEL", "MIGUEL", "ALEXIS"};
char Apellido1[20][10] = {"SANCHEZ", "GONZALEZ", "RAMIREZ", "LOPEZ", "SORIANO", "ESCOBAR", "FERNANDES", "VARGAS", "ALMARAS", "PEREZ", "MENDOZA"};
char Apellido2[20][10] = {"MARTINEZ", "PEREZ", "SILVA", "PALACIOS", "DIAZ", "ALDANA", "AGUILAR", "RODRIGUEZ", "SOTO", "RIOS", "CASTRO", "MORALI"};

int main()
{
    srand(time(NULL));
    menu();
    return 0;
}

//
int msges()
{
    int op;
    system("CLS");
    printf("\n MOD MENU \n");
    printf("1.- AGREGAR AUTOMATICO \n");
    printf("2.- AGREGAR MANUAL \n");
    printf("3.- ELIMINAR REGISTRO \n");
    printf("4.- BUSCAR \n");
    printf("5.- ORDENAR \n");
    printf("6.- IMPRIMIR \n");
    printf("0.- SALIR \n");
    op = ValidarCadena("Escribe el numero de opcion: \n", 0, 6);

    return op;
}

void menu()

```

A vertical toolbar with a blue button and a scroll bar.

```

    }
    else
    {
        printf("El alumno con matricula %d ya ha sido eliminado anteriormente.\n", EliminarMatricula);
    }
    encontrado = 1;
}

if (!encontrado)
{
    printf("Alumno con matricula %d no encontrado.\n", EliminarMatricula);
}
TipoB = 0;
system("PAUSE");
break;

case 4:
    if(TipoB == 0)
    {
        MatriculaBuscada = ValidarCadena("Ingresa la matricula (entre 300000 y 399999): ", 300000, 399999);
        indice = BuscarMatricula(alumno, MatriculaBuscada, i);
        if (indice != -1)
        {
            printf("| %-7s | %-12s | %-40s | %-6s | %-9s |\n", "STATUS", "MATRICULA", "NOMBRE", "EDAD", "SEXO");

            if (alumno[indice].status == 1)
            {
                printf("| %-7d | %-12d | %-40s | %-6d | %-9s |\n", alumno[indice].status, alumno[indice].matricula, alumno[indice].
            }
        }
    }
}

```

```

    for (int j = 0; j < i; j++)
    {
        if (alumno[j].status == 1)
        {
            printf("| %-7d | %-12d | %-40s | %-6d | %-9s |\n", alumno[j].status, alumno[j].matricula, alumno[j].nombre, alumno[j].
        }
    }

    system("PAUSE");
    break;

case 6:

    printf("| %-7s | %-12s | %-40s | %-6s | %-9s |\n", "STATUS", "MATRICULA", "NOMBRE", "EDAD", "SEXO");

    for (int j = 0; j < i; j++)
    {
        if (alumno[j].status == 1)
        {
            printf("| %-7d | %-12d | %-40s | %-6d | %-9s |\n", alumno[j].status, alumno[j].matricula, alumno[j].nombre, alumno[j].
        }
    }

    TipoB = 0;
    system("PAUSE");
    break;

case 0:
    printf("Saliendo del programa.\n");
    break;

```

```

        default:
            printf("Opcion no valida.\n");
            break;
    }
} while (op != 0);
}

Talumno GenerarAutom(void)
{
    Talumno alumno;

    alumno.status = 1;
    alumno.matricula = GenerarMatriculaRandom();

    char sexo = GenerarNombreRandom(alumno.nombre);
    alumno.edad = GenerarEdadRandom();
    alumno.sexo = sexo;

    return alumno;
}

int rango(int ri, int rf) {
    return ri + rand() % (rf - ri + 1);
}

char GenerarNombreRandom(char nombre[])
{
    char PriNom[10];

```

```

    char PriNom[10];
    char SegNom[10];
    char PriApe[10];
    char SegApe[10];
    char sexo;

    if (rand() % 2 == 0)
    {
        sexo = 0;
        strcpy(PriNom, PriNomFem[rango(0, 9)]);
        if (rand() % 2 == 0)
        {
            strcpy(SegNom, SegNomFem[rango(0, 9)]);
        }
        else
        {
            SegNom[0] = '\0';
        }
    }
    else
    {
        sexo = 1;
        strcpy(PriNom, PriNomMas[rango(0, 9)]);
        if (rand() % 2 == 0)
        {
            strcpy(SegNom, SegNomMas[rango(0, 9)]);
        }
        else
        {
            SegNom[0] = '\0';
        }
    }
}

```

```

    }

strcpy(PriApe, Apellido1[rango(0, 19)]);
strcpy(SegApe, Apellido2[rango(0, 19)]);

nombre[0] = '\0';
strcat(nombre, PriNom);
if (SegNom[0] != '\0')
{
    strcat(nombre, " ");
    strcat(nombre, SegNom);
}

strcat(nombre, " ");
strcat(nombre, PriApe);
strcat(nombre, " ");
strcat(nombre, SegApe);

return sexo;
}

int GenerarMatriculaRandom()
{
    int ri = 300000;
    int rf = 399999;
    return ri + rand() % (rf - ri + 1);
}

```

```

int GenerarEdadRandom()
{
    int ri = 17;
    int rf = 59;
    return ri + rand() % (rf - ri + 1);
}

Talumno GenerarManual(void)
{
    Talumno alumno;
    char PriNom[100];
    char SegNom[100];
    char PriApe[100];
    char SegApe[100];
    int op;

    alumno.status = 1;

    alumno.matricula = ValidarCadena("Ingresa la matricula (entre 300000 y 399999): ", 300000, 399999);

    do {
        printf("Ingresa tu primer nombre: \n");
        fflush(stdin);
        gets(PriNom);
        Mayusculas(PriNom);
        op = ValidarCadenaTexto(PriNom);
        if (op != 1)
        {
            printf("Nombre no valido.\n");
        }
    } while (op != 1);
}

```

```

strcpy(alumno.nombre, PriNom);

op = ValidarCadena("Tiene un segundo nombre? (SI = 0, NO = 1): \n", 0, 1);
if (op == 0)
{
    printf("Ingresa tu segundo nombre: \n");
    fflush(stdin);
    gets(SegNom);
    Mayusculas(SegNom);
    op = ValidarCadenaTexto(SegNom);
    if (op != 1)
    {
        printf("Nombre no valido.\n");
    }
    strcat(alumno.nombre, " ");
    strcat(alumno.nombre, SegNom);
}

do {
    printf("Ingresa tu primer apellido: \n");
    fflush(stdin);
    gets(PriApe);
    Mayusculas(PriApe);
    op = ValidarCadenaTexto(PriApe);
    if (op != 1)
    {
        printf("Apellido no valido.\n");
    }
} while (op != 1);

op = ValidarCadena("Tiene un segundo apellido? (SI = 0, NO = 1): \n", 0, 1);

```

```

if (op == 0)
{
    printf("Ingresa tu segundo apellido: \n");
    fflush(stdin);
    gets(SegApe);
    Mayusculas(SegApe);
    op = ValidarCadenaTexto(SegApe);
    if (op != 1)
    {
        printf("Apellido no valido.\n");
    }

    strcat(alumno.nombre, " ");
    strcat(alumno.nombre, PriApe);
    strcat(alumno.nombre, " ");
    strcat(alumno.nombre, SegApe);
}

alumno.edad = ValidarCadena("Ingresa la edad (entre 17 y 59): ", 17, 59);

alumno.sexo = ValidarCadena("Ingresa el sexo (0 para Femenino o 1 para Masculino): ", 0, 1);

return alumno;
}

int BuscarMatricula(Talumno alumnos[], int BuscarMatricula, int cantidad)
{
    int i;
    for (i = 0; i < cantidad; i++)
    {
        if (alumnos[i].status == 1)

```

```

        if(alumnos[i].matricula == BuscarMatricula)
        {
            return i;
        }
    }
    return -1;
}

int BusquedaBinaria(Talumno alumnos[], int i, int d, int Matricula)
{
    while (i <= d)
    {
        int m = i + (d - i) / 2;

        if (alumnos[m].matricula == Matricula)
        {
            return m;
        }

        if (alumnos[m].matricula < Matricula)
        {
            i = m + 1;
        }

        if (alumnos[m].matricula > Matricula)
        {
            d = m - 1;
        }
    }
}

```

```

}

int OrdenarMatricula(Talumno alumnos[], int cantidad)
{
    int i, j;
    Talumno temp;

    for (i = 0; i < cantidad - 1; i++)
    {
        for (j = 0; j < cantidad - i - 1; j++)
        {
            if (alumnos[j].status == 1)
            {
                if(alumnos[j + 1].status == 1)
                {
                    if(alumnos[j].matricula > alumnos[j + 1].matricula)
                    {
                        temp = alumnos[j];
                        alumnos[j] = alumnos[j + 1];
                        alumnos[j + 1] = temp;
                    }
                }
            }
        }
    }
    return -1;
}

```


