



```

        break;
        case 1:
            printf("\nSOLO HAY UN ALUMNO\n");
            break;
        default:
            ordenarAlum(alumnos,i);
    }
    break;
case 6:
    if(i>0)
    {
        printf("| No | Status | Matricula | Apellido Paterno | Apellido Materno | Nombre | Edad | Sexo | \n");
        for(j=0;j<i;j++)
        {
            printAlum(alumnos,j);
        }
    }
    else
    {
        printf("\nAUN NO HAY ALUMNOS\n");
    }
    break;
case 7:
    printf("\nSALIENDO DEL PROGRAMA\n");
    break;
}
system("PAUSE");
}while(op!=7);
}

int buscarAlum(StructAlum alumnos[],int i,int num)
{
    int j;
    for(j=0;j<i;j++) //Recorre el vector de alumnos
    {
        if(num==alumnos[j].matricula) //Compara la matricula buscada con las existentes
        {
            return j; //Si la encuentra regresa la posicion
        }
    }
    return -1; //Si no la encontro regresa -1
}

int genMatri(StructAlum alumnos[],int i)
{
    int num;
    do
    {
        num=(rand() %100000)+300000;
    }while(buscarAlum(alumnos,i,num)!=-1);
    return num;
}

void strConcat(char str2[],char str1[])
{
    int j,k=0;
    j=strlen(str2); //Calcula la longitud de la cadena principal
    str2[j++]=' '; //Cambia el caracter nulo por espacio
    while(str1[k]!='\0') //Se repite hasta que la cadena que se concatenara llegue al caracter nulo
    {
        str2[j]=str1[k]; //Despues del espacio se le asigna la segunda cadena a la primera
        k++;
        j++;
    }
    str2[j]='\0'; //Se asigna el caracter nulo despues de haber concatenado
}

void genH_Name(StructAlum alumnos[],int i)
{
    if(rand() %2)
    {
        strcpy(alumnos[i].nombre,m_name[rand() %28]); //Genera un nombre de la libria que contiene nombres
    }
    else
    {
        strcpy(alumnos[i].nombre,m_name[rand() %28]); //Genera un nombre de la libria que contiene nombres
        strConcat(alumnos[i].nombre,m_name[rand() %28]); //Concatena el primer nombre con otro generado de la libreria
    }
}

void genM_Name(StructAlum alumnos[],int i)
{
    if(rand() %2)
    {
        strcpy(alumnos[i].nombre,m_name[rand() %28]); //Genera un nombre de la libria que contiene nombres
    }
    else
    {
        strcpy(alumnos[i].nombre,m_name[rand() %28]); //Genera un nombre de la libria que contiene nombres
        strConcat(alumnos[i].nombre,m_name[rand() %28]); //Concatena el primer nombre con otro generado de la libreria
    }
}

void randAlum(StructAlum alumnos[],int i)
{
    int prob; //se usa para controlar la probabilidad de tener 1 o 2 apellidos
    alumnos[i].status=rand() %2;
    alumnos[i].matricula=genMatri(alumnos,i);
    alumnos[i].sexo=(rand() %2)+1;
    alumnos[i].edad=(rand() %150)+1;
    if(alumnos[i].sexo==1)
    {
        genH_Name(alumnos,i);
    }
    else
    {
        genM_Name(alumnos,i);
    }

    prob=rand() %10;
    switch(prob)
    {
        case 0:
        case 1:
        case 2:
        case 3:
        case 4:
    }
}

```

```

        case 5:
        case 6:
            strcpy(alumnos[i].apPat,ap[rand()%48]);
            strcpy(alumnos[i].apMat,ap[rand()%48]);
            break;
        case 7:
        case 8:
        case 9:
            strcpy(alumnos[i].apPat,ap[rand()%48]);
            strcpy(alumnos[i].apMat," ");
            break;
    }
}

void scanAlum(StructAlum alumnos[],int i)
{
    int matri,validmatri;
    alumnos[i].status=validNum("Ingresa su status (0 o 1): ",0,1);
    do
    {
        matri=validNum("Ingresa su matricula: ",300000,399999);
        validmatri=buscarAlum(alumnos,i,matri);
        if(validmatri!=-1)
        {
            printf("\nEsa matricula ya existe\n");
        }
    }while(validmatri!=-1);
    alumnos[i].matricula=matri;
    printf("\nIngresa su apellido paterno: ");
    matri=validStr(alumnos[i].apPat);
    strM(alumnos[i].apPat);
    printf("\nIngresa su apellido materno: ");
    matri=validStr(alumnos[i].apMat);
    strM(alumnos[i].apMat);
    printf("\nIngresa su nombre: ");
    matri=validStr(alumnos[i].nombre);
    strM(alumnos[i].nombre);
    alumnos[i].edad=validNum("\nIngresa su edad ",1,150);
    alumnos[i].sexo=validNum("\nIngresa su sexo\n1) Hombre\n2) Mujer\nEscribe una opcion: ",1,2);
}

void delAlum(StructAlum alumnos[],int i,int pos)
{
    int j;
    for(j=pos;j<i;j++) //Recorre las posiciones de los alumnos
    {
        alumnos[j]=alumnos[j+1]; //La posiciÃn del alumno actual es remplazada por el siguiente alumno
    }
}

void printAlum(StructAlum alumnos[],int j)
{
    printf("| %3d | %d | %9d | %-16s | %-17s | %-19s | %4d |",j+1,alumnos[j].status,alumnos[j].matricula,alumnos[j].apPat,alumnos[j].apMat,alumnos[j].nombre,alumnos[j].edad);
    if(alumnos[j].sexo==1)
    {
        printf(" HOMBRE | \n");
    }
    else
    {
        printf(" MUJER | \n");
    }
}

void ordenarAlum(StructAlum alumnos[],int n)
{
    int i,j,swap;
    StructAlum temp; //guarda la posicion del alumno para intercambiarla con otro
    for(i=0;i<n-1;i++)
    {
        swap=0; //Se asigna 0 a la bandera de intercambios
        for(j=0;j<n-i-1;j++)
        {
            if(alumnos[j].matricula>alumnos[j+1].matricula) //Si la matricula siguiente es menor a la actual se intercambian
            {
                temp=alumnos[j]; //guarda los datos del alumno actual
                alumnos[j]=alumnos[j+1]; //asigna los datos del siguiente alumno al actual
                alumnos[j+1]=temp; //asigna los datos guardados del primer alumno al segundo alumno
                swap=1; //asigna 1 a la bandera de intercambios
            }
        }
        if (swap==0)
        {
            i=n; //Si no se hicieron intercambios i se iguala a n para que termine el ciclo
        }
    }
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NM 4
#define STR 100

/** Validacion *****/

//Valida un numero entero
//Parametros:
//-txt Cadena a mostrar antes de leer la varia
//-limi Limite inferior aceptado
//-lims Limite superior aceptado
int validNum(char txt[],int limi,int lims);

//Valida una cadena
//Parametros:
//-str Cadena a validar
//Devuelve:
//-invalido Numero de errores
//Solo permite letras
//No permite empezar ni terminar con espacios
//No permite dobles pesacios
//Muestra los errores
int validStr(char str[]);

/** Cadenas *****/

//Convierte una cadena a mayusculas
//Parametros:
//-str Cadena a convertir a mayusculas
void strM(char str[]);

//Convierte una cadena a minusculas
//Parametros:
//-str Cadena a convertir a minusculas
void strm(char str[]);

//Regresa la longitud de la cadena
//Parametros:
//-str Cadena de la que se quiere saber su longitud
int strlen(char str[]);

//Imprime una cadena
//Parametros:
//-str Cadena que se quiere imprimir
void strprint(char str[]);

//Copia la cadena en la cadena 2
//Parametros:
//-str Cadena original
//-str2 Cadena que en la que se copiara str
void strcpy(char str2[],char str[]);

//Valida un numero entero
//Parametros:
//-txt Texto a mostrar al leer el numero
//Devuelve:
//Numero validado
//No requiere un limite para funcionar
int validNumUnLimit(char txt[]);

/** Arreglos *****/

/** Asignacion **
//Regresan 1, indica que se lleno

//Llena un vector
//Parametros:
//vect Vector a llenar
//n Tamaño
//limi Limite inferior aceptable
//lims Limite superior aceptable
int scanVector(int vect[],int n,int limi,int lims);

```

```

//Llena un vector con numeros aleatorios sin repetir
//Parametros:
//-vect    Vector a llenar
//-n        Tamaño
//-ri       Rango inicial
//-rf       Rango final
int randVector(int vect[],int n,int ri,int rf);

//Llena una matriz
//Parametros:
//-matriz   Matriz a llenar
//-r        Numero de renglones
//-c        Numero de columnas
//-limi     Limite inferior aceptable
//-lims     Limite superior aceptable
int scanMatriz(int matriz[][NM],int r,int c,int limi,int lims);

//Llena una matriz con numeros aleatorios sin repetir
//Parametros:
//-matriz   Matriz a llenar
//-r        Numero de renglones
//-c        Numero de columnas
//-ri       Rango inicial
//-rf       Rango final
int randMatriz(int matriz[][NM],int r,int c,int ri,int rf);

//Imprimir vector
//Parametros:
//-vect     Vector a imprimir
//-n        Tamaño
void printVector(int vect[],int n);

//Imprimir matriz
//Parametros:
//-matriz   Matriz a imprimir
//-r        Numero de renglones
//-c        Numero de columnas
void printMatriz(int matriz[][NM],int r,int c);

//Busca un valor en el vector
//Parametros:
//-vect     Vector en el que se buscara
//-n        Tamaño
//-num      Numero a buscar
//Devuelve:
//Posicion si encontro el numero
//-1 Si no encontro el numero
int searchVector(int vect[],int n,int num);

//Busca un valor en una matriz
//Parametros:
//-matriz   Matriz en la que se buscara
//-pos      Guarda la posicion en i y j
//-r        Numero de renglones
//-c        Numero de columnas
//-num      Numero a buscar
//Devuelve:
//Renglon en que se encontro el numero
//-1 Si no encontro el numero
int searchMatriz(int matriz[][NM],int pos[],int r,int c, int num);

/** Metodos de ordenacion *****/

void improvedBubbleSort(int vect[],int n);

int validNum(char txt[],int limi,int lims)
{
    char r[STR];
    int rint;
    do
    {
        printf("%s",txt);
        fflush(stdin);
        gets(r);
        rint=atoi(r);
        if(rint<limi || rint>lims)

```

```

        {
            printf("Valor invalido, ingresalo de nuevo\n");
        }
    }while(rint<limi || rint>lims);
    return rint;
}

int validStr(char str[])
{
    int i=0,invalido=0;
    fflush(stdin);
    gets(str);
    do
    {
        if(str[i]>90)
        {
            if(str[i]<97)
            {
                printf("\nEl simbolo %c no es valido \n",str[i]);
                invalido++;
            }
        }
        else
        {
            if(str[i]>122)
            {
                printf("\nEl simbolo %c no es valido \n",str[i]);
                invalido++;
            }
            else
            {
                if(str[i]<65)
                {
                    if(str[i]==' ')
                    {
                        if(i==0)
                        {
                            printf("\nLa cadena no debe empezar con espacio\n");
                            invalido++;
                        }
                        else
                        {
                            if(str[i-1]==' ')
                            {
                                printf("\nLa cadena no debe llevar dos espacios seguidos\n");
                                invalido++;
                            }
                        }
                    }
                    else
                    {
                        if(str[i]!='\0')
                        {
                            if(str[i-1]==' ')
                            {
                                printf("\nLa cadena no debe terminar en espacio\n");
                                invalido++;
                            }
                        }
                        else
                        {
                            printf("\nEl simbolo %c no es valido \n",str[i]);
                            invalido++;
                        }
                    }
                }
            }
        }
    }
    i++;
    }while(str[i-1]!='\0');
    return invalido;
}

int validNumUnLimit(char txt[])
{
    char r[STR];

```

```

    int rint;
    printf("%s",txt);
    fflush(stdin);
    gets(r);
    rint=atoi(r);
    return rint;
}

void strM(char str[])
{
    int i;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]>=97)
        {
            if(str[i]<=122)
            {
                str[i]-=32;
            }
        }
    }
}

void strm(char str[])
{
    int i;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]>=65)
        {
            if(str[i]<=90)
            {
                str[i]+=32;
            }
        }
    }
}

int strlen(char str[])
{
    int i=0;
    while(str[i]!='\0')
    {
        i++;
    }
    return i;
}

void strprint(char str[])
{
    int i;
    for(i=0;str[i]!='\0';i++)
    {
        printf("%c",str[i]);
    }
}

void strcpy(char str2[],char str[])
{
    int i=0;
    for(i=0;str[i]!='\0';i++)
    {
        str2[i]=str[i];
    }
    str2[i]='\0';
}

int scanVector(int vect[],int n,int limi,int lims)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("Ingresa el valor de la posicion: %d\n",i+1);
        vect[i]=validNum("",limi,lims);
    }
    return 1;
}

```

```

}

int randVector(int vect[],int n,int ri,int rf)
{
    int i,num,range;
    srand(time(NULL));
    range=(rf-ri)+1;
    for(i=0;i<n;i++)
    {
        do
        {
            num=(rand()%range)+ri;
        }while(searchVector(vect,n,num)!=-1);
        vect[i]=num;
    }
    return 1;
}

int scanMatriz(int matriz[][NM],int r,int c,int limi,int lims)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("Ingresa el valor de la posicion: [%d][%d]\n",i+1,j+1);
            matriz[i][j]=validNum("",limi,lims);
        }
    }
    return 1;
}

int randMatriz(int matriz[][NM],int r,int c,int ri,int rf)
{
    int i,j,k;
    int n;
    n=r*c;
    int vect[n];
    randVector(vect,n,ri,rf);
    k=0;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            matriz[i][j]=vect[k];
            k++;
        }
    }
    return 1;
}

void printVector(int vect[],int n)
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("%d ",vect[i]);
    }
}

void printMatriz(int matriz[][NM],int r,int c)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            printf("%4d ",matriz[i][j]);
        }
        printf("\n");
    }
}

int searchVector(int vect[],int n,int num)
{
    int i;

```



```

for(i=0;i<n;i++)
{
    if(vect[i]==num)
    {
        return i;
    }
}
return -1;
}

int searchMatriz(int matriz[][NM],int pos[],int r,int c,int num)
{
    int i,j;
    for(i=0;i<r;i++)
    {
        for(j=0;j<c;j++)
        {
            if(matriz[i][j]==num)
            {
                pos[0]=i;
                pos[1]=j;
                return i;
            }
        }
    }
    return -1;
}

void improvedBubbleSort(int vect[],int n)
{
    int i,j;
    int temp,swap;
    for(i=0;i<n-1;i++)
    {
        swap=0;
        for(j=0;j<n-i-1;j++)
        {
            if(vect[j]>vect[j+1])
            {
                temp=vect[j];
                vect[j]=vect[j+1];
                vect[j+1]=temp;
                swap=1;
            }
        }
        if (swap==0)
        {
            i=n;
        }
    }
}

```

```
#include <stdio.h>

//30 Nombres
char h_name[30][30]={
"JOSE","JUAN","LUIS","MANUEL","ANTONIO","JESUS","CARLOS","FRANCISCO","ALBERTO","JORGE","MIGUEL",
"ANGEL","JAVIER","ALEJANDRO","ENRIQUE","VICTOR","ARTURO","CESAR","FERNANDO","PEDRO","MARTIN",
"ROBERTO","EDUARDO","MARIO","ARMANDO","SERGIO","RAUL","ALFREDO","RAFAEL","RICARDO"};

//30 Nombres
char m_name[30][30]={
"MARIA","GUADALUPE","ROSA","MARTHA","ANA","PATRICIA","LETICIA","ELENA","LAURA","ELIZABETH","ISABEL",
"ALICIA","MARGARITA","LUZ","ANGELICA","CLAUDIA","SILVIA","JUANA","GABRIELA","ADRIANA","VERONICA",
"TERESA","ALEJANDRA","ALMA","BEATRIZ","ARACELI","YOLANDA","BLANCA","SANDRA"};

char ap[50][30]={
"HERNANDEZ","GARCIA","MARTINEZ","LOPEZ","GONZALEZ","RODRIGUEZ","PEREZ","SANCHEZ","RAMIREZ","FLORES","CRUZ",
"GOMEZ","MORALES","VAZQUEZ","REYES","DIAZ","TORRES","GUTIERREZ","RUIZ","AGUILAR","MENDOZA","CASTILLO","ORTIZ",
"MORENO","RIVERA","RAMOS","ROMERO","JUAREZ","ALVAREZ","MENDEZ","CHAVEZ","HERRERA","MEDINA","DOMINGUEZ",
"CASTRO","GUZMAN","VARGAS","VELAZQUEZ","SALAZAR","ROJAS","ORTEGA","CORTES","SANTIAGO","GUERRERO","CONTRERAS",
"BAUTISTA","ESTRADA","LUNA","LARA","VASQUEZ"};
```

## Pruebas de escritorio:

### Menú:

```
MENU
1) AGREGAR AUTOMATICO
2) AGREGAR MANUAL
3) ELIMINAR REGISTRO
4) BUSCAR
5) ORDENAR
6) IMPRIMIR
7) SALIR
ESCOGE UNA OPCION: █
```

### Opción 1:

```
SE GENERARON 10 ALUMNOS
Presione una tecla para continuar . . . █
```

### Opción 2:

```
AGREGAR UN ALUMNO

Ingresa su status (0 o 1): 1
Ingresa su matricula: 372099

Ingresa su apellido paterno: MEJIA

Ingresa su apellido materno: MARTINEZ

Ingresa su nombre: RAMON

Ingresa su edad 20

Ingresa su sexo
1) Hombre
2) Mujer
Escoge una opcion: 1
Presione una tecla para continuar . . . █
```

### Opción 3:

```
ELIMINAR ALUMNO

INGRESA LA MATRICULA DEL ALUMNO: 372099

SE ELIMINO EL ALUMNO:
| No | Status | Matricula | Apellido Paterno | Apellido Materno | Nombre | Edad | Sexo |
| 11 | 1 | 372099 | MEJIA | MARTINEZ | RAMON | 20 | Hombre |
Presione una tecla para continuar . . .
```

### Opción 4:

```
BUSCAR ALUMNO

INGRESA LA MATRICULA DEL ALUMNO: 322411
| No | Status | Matricula | Apellido Paterno | Apellido Materno | Nombre | Edad | Sexo |
| 4 | 0 | 322411 | ROJAS | ESTRADA | BEATRIZ | 19 | Hombre |
Presione una tecla para continuar . . .
```

### Opción 5:

```
SE ORDENARON LOS ALUMNOS POR MATRICIAL

Presione una tecla para continuar . . .
```

```
| No | Status | Matricula | Apellido Paterno | Apellido Materno | Nombre | Edad | Sexo |
| 1 | 0 | 300142 | MEDINA | VELAZQUEZ | VERONICA BEATRIZ | 5 | Mujer |
| 2 | 1 | 302549 | ALVAREZ | LOPEZ | TERESA GABRIELA | 50 | Mujer |
| 3 | 1 | 310844 | RIVERA | HERRERA | ELIZABETH | 38 | Mujer |
| 4 | 1 | 314735 | RAMOS | GOMEZ | GABRIELA ALICIA | 36 | Hombre |
| 5 | 1 | 322151 | BAUTISTA | HERNANDEZ | LETICIA SILVIA | 66 | Hombre |
| 6 | 0 | 322411 | ROJAS | ESTRADA | BEATRIZ | 19 | Hombre |
| 7 | 0 | 325729 | LUNA | CONTRERAS | BLANCA ISABEL | 71 | Hombre |
| 8 | 1 | 326823 | CHAVEZ | GOMEZ | ALICIA ELIZABETH | 62 | Mujer |
| 9 | 0 | 328956 | FLORES | GONZALEZ | ALEJANDRA | 124 | Hombre |
| 10 | 1 | 329247 | CONTRERAS | | LETICIA CLAUDIA | 112 | Hombre |
Presione una tecla para continuar . . .
```

## Opción 6:

No	Status	Matricula	Apellido Paterno	Apellido Materno	Nombre	Edad	Sexo
1	0	300142	MEDINA	VELAZQUEZ	VERONICA BEATRIZ	5	Mujer
2	1	302549	ALVAREZ	LOPEZ	TERESA GABRIELA	50	Mujer
3	1	310844	RIVERA	HERRERA	ELIZABETH	38	Mujer
4	1	314735	RAMOS	GOMEZ	GABRIELA ALICIA	36	Hombre
5	1	322151	BAUTISTA	HERNANDEZ	LETICIA SILVIA	66	Hombre
6	0	322411	ROJAS	ESTRADA	BEATRIZ	19	Hombre
7	0	325729	LUNA	CONTRERAS	BLANCA ISABEL	71	Hombre
8	1	326823	CHAVEZ	GOMEZ	ALICIA ELIZABETH	62	Mujer
9	0	328956	FLORES	GONZALEZ	ALEJANDRA	124	Hombre
10	1	329247	CONTRERAS		LETICIA CLAUDIA	112	Hombre
11	1	300868	RAMIREZ	SALAZAR	JUANA	144	Mujer
12	1	313901	MORALES	MORENO	BEATRIZ	24	Mujer
13	0	316504	LOPEZ	GUTIERREZ	ISABEL MARIA	131	Mujer
14	0	328890	DOMINGUEZ	MENDOZA	MARGARITA	30	Mujer
15	0	324740	SALAZAR	GUZMAN	MARTHA	91	Hombre
16	1	324286	HERRERA		ALICIA	126	Hombre
17	0	300267	VELAZQUEZ	FLORES	ADRIANA MARTHA	79	Mujer
18	0	318011	GUERRERO	DOMINGUEZ	ANGELICA	91	Hombre
19	0	313448	PEREZ	VELAZQUEZ	JUANA ROSA	118	Hombre
20	0	312850	HERRERA	LUNA	ARACELI VERONICA	88	Hombre

Presione una tecla para continuar . . .