



Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Lizeth Barrios Retana

Matrícula: 372813

Maestro: Pedro Núñez Yépiz

Actividad No. : 10 CÓDIGO Y CAPTURA DE PANTALLAS

**Tema - Unidad : Funciones y Metodos de ordenacion y Busqueda,
Estructuras y Librerías**

Ensenada, Baja California a 15 de octubre de 2023.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

• Código principal

```
/* Lizeth Barrios Retana 372813
10-Oct-2023
REALICE EL SIGUIENTE PROGRAMA QUE CONTENGA UN MENÚ.

    MENÚ
    1.- AGREGAR (AUTOM 10 REGISTROS)
    2.- AGREGAR MANUAL
    3.- ELIMINAR REGISTRO (lógico)
    4.- BUSCAR
    5.- ORDENAR
    6.- IMPRIMIR
    0.- SALIR

UTILIZAR UN ARREGLO DE 500 REGISTROS
SE DEBERÁ UTILIZAR ESTRUCTURAS CON LOS DATOS BÁSICOS DE UN ALUMNO ( status,
Matricula, ApPat,
ApMat, Nombre, Edad, Sexo )
Busqueda y Ordenacion por campo MATRICULA
BRL_Act10_01_932
*/
#include <stdio.h>
#include <string.h>
#include <time.h>
#include "Lizeth.h"

struct InfoAlumno
{
    char Nombre[20];
    char ApMat[20];
    char ApPat[20];
    int matricula;
    char estado[20];
    int edad;
    char sexo[10];
};

char hombres[20][10] = {"JUAN", "PEDRO", "CARLOS", "LUIS", "JAVIER", "ANDRES",
"ALEJANDRO", "MIGUEL", "JOSE", "ADAN", "JORGE", "GABRIEL", "ROBERTO", "DAVID",
"FERNANDO", "RAUL", "EDUARDO", "SERGIO", "ANTONIO", "FRANCISCO"};

char mujeres[20][10] = {"DANIELA", "LAURA", "ANA", "CARMEN", "SOFIA", "ISABEL",
"MARTA", "PAULA", "ELENA", "PATRICIA", "LUCIA", "CLARA", "ROSA", "BEATRIZ",
"CAROLINA", "JULIA", "TERESA", "SARA", "IRENE", "ADRIANA"};

char ApMat[20][10] = {"PEREZ", "GONZALEZ", "RODRIGUEZ", "LOPEZ", "MARTINEZ",
"SANCHEZ", "RAMIREZ", "TORRES", "FLORES", "GARCIA", "DIAZ", "FERNANDEZ", "RUIZ",
"CASTRO", "MORALES", "ORTIZ", "HERRERA", "BARRIOS", "JIMENEZ", "SILVA"};

char ApPat[20][10] = {"CRUZ", "LOZANO", "ROJAS", "REYES", "MENDOZA", "GUZMAN",
"ROMERO", "VASQUEZ", "MUNOZ", "GOMEZ", "LUNA", "SANTOS", "CUEVAS", "CHAVEZ",
"MEDINA", "DELGADO", "GALLEGOS", "MIRANDA", "RETANA", "RIOS"};

char stat[4][20] = {"DADO DE BAJA", "BAJA TEMPORAL", "MATRICULADO", "EGRESADO"};
char sexo[2][10] = {"HOMBRE", "MUJER"};
int genero, nombres;

void agregarAutom(struct InfoAlumno estudiantes[], int n);
void imprimir(struct InfoAlumno estudiantes[], int numEstudiantes);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
void agregarManual(struct InfoAlumno estudiantes[], int *numEstudiantes);
void eliminarRegistro(struct InfoAlumno estudiantes[], int *numEstudiantes);
void buscar(struct InfoAlumno estudiantes[], int *numEstudiantes);
void ordenar(struct InfoAlumno estudiantes[], int numEstudiantes);

int main()
{
    srand(time(NULL));
    struct InfoAlumno estudiantes[500];
    int opcion, menu, i;
    int numEstudiantes = 0;
    do
    {
        menu = 0;

        opcion = validarNumeros("\n===MENU===\n1.- AGREGAR\n2.- AGREGAR MANUAL\n3.-
ELIMINAR REGISTRO (logico)\n4.- BUSCAR\n5.- ORDENAR\n6.- IMPRIMIR\n0.- SALIR\n", 0,
6);

        switch (opcion)
        {
            case 1:
                for (i = numEstudiantes; i < numEstudiantes + 10; i++)
                {
                    agregarAutom(estudiantes, i);
                }
                numEstudiantes += 10;
                menu = repetir();
                break;
            case 2:
                agregarManual(estudiantes, &numEstudiantes);
                menu = repetir();
                break;
            case 3:
                eliminarRegistro(estudiantes, &numEstudiantes);
                menu = repetir();
                break;
            case 4:
                buscar(estudiantes, &numEstudiantes);
                menu = repetir();
                break;
            case 5:
                ordenar(estudiantes, numEstudiantes);
                menu = repetir();
                break;
            case 6:
                imprimir(estudiantes, numEstudiantes);
                menu = repetir();
                break;
            case 0:
                printf("FIN DEL PROGRAMA!\n");
                return 0;
        }
    } while (menu == 1);

    return 0;
}

void agregarAutom(struct InfoAlumno estudiantes[], int n)
{
    strcpy(estudiantes[n].ApPat, ApPat[random(1, 20)]);
    strcpy(estudiantes[n].ApMat, ApMat[random(1, 20)]);
}
```



```
estudiantes[n].matricula = random(100000, 500000);
strcpy(estudiantes[n].estado, stat[random(0, 3)]);
estudiantes[n].edad = random(18, 40);
genero = random(0, 1);
nombres = random(1, 2);

if (nombres == 1)
{
    if (genero == 0)
    {
        strcpy(estudiantes[n].sexo, "HOMBRE");
        strcpy(estudiantes[n].Nombre, hombres[random(0, 19)]);
    }
    else
    {
        strcpy(estudiantes[n].sexo, "MUJER");
        strcpy(estudiantes[n].Nombre, mujeres[random(0, 19)]);
    }
}
else
{
    if (genero == 0)
    {
        strcpy(estudiantes[n].sexo, "HOMBRE");
        strcpy(estudiantes[n].Nombre, hombres[random(0, 19)]);
        strcat(estudiantes[n].Nombre, " ");
        strcat(estudiantes[n].Nombre, hombres[random(0, 19)]);
    }
    else
    {
        strcpy(estudiantes[n].sexo, "MUJER");
        strcpy(estudiantes[n].Nombre, mujeres[random(0, 19)]);
        strcat(estudiantes[n].Nombre, " ");
        strcat(estudiantes[n].Nombre, mujeres[random(0, 19)]);
    }
}
}

void agregarManual(struct InfoAlumno estudiantes[], int *numEstudiantes)
{
    if (*numEstudiantes >= 500)
    {
        printf("EL LIMITE DEL REGISTRO HA SIDO ALCANZADO\n");
        return;
    }

    struct InfoAlumno *persona = &estudiantes[*numEstudiantes];
    int status, matricula, edad, genero;
    char ApPat[20], ApMat[20], segundoNombre[20];
    nombres = validarNumeros("TIENES 1 O 2 NOMBRES?\n", 1, 2);
    if (nombres == 2)
    {
        validarLetras("INGRESE EL PRIMER NOMBRE: ", persona->Nombre,
sizeof(persona->Nombre));
        validarLetras("INGRESE EL SEGUNDO NOMBRE: ", segundoNombre,
sizeof(segundoNombre));
        strcat(persona->Nombre, " ");
        strcat(persona->Nombre, segundoNombre);
    }
    else
    {

```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        validarLetras("INGRESE EL NOMBRE: ", persona->Nombre,
sizeof(persona->Nombre));
    }
    validarLetras("INGRESE EL APELLIDO PATERNO: ", persona->ApPat,
sizeof(persona->ApPat));
    validarLetras("INGRESE EL APELLIDO MATERNO: ", persona->ApMat,
sizeof(persona->ApMat));
    matricula = validarNumeros("INGRESE LA MATRICULA: ", 100000, 500000);
    persona->matricula = matricula;
    edad = validarNumeros("INGRESE LA EDAD: ", 18, 30);
    persona->edad = edad;
    genero = validarNumeros("INGRESE EL NUMERO DEL GÉNERO:\n0.- HOMBRE\n1.-
MUJER\n", 0, 1);
    strcpy(persona->sexo, sexo[genero]);
    status = validarNumeros("CUAL ES EL ESTADO ACADEMICO?\n0.- DADO DE BAJA\n1.-
BAJA TEMPORAL\n2.- MATRICULADO\n3.- EGRESADO\n", 0, 3);
    strcpy(persona->estado, stat[status]);
    (*numEstudiantes)++;
}

void eliminarRegistro(struct InfoAlumno estudiantes[], int *numEstudiantes)
{
    int i, j, matricula, encontrado = 0;

    matricula = validarNumeros("INGRESE LA MATRICULA DEL ALUMNO PARA ELIMINAR SU
REGISTRO: ", 100000, 500000);

    for (i = 0; i <= *numEstudiantes; i++)
    {
        if (estudiantes[i].matricula == matricula)
        {
            for (j = i; j < *numEstudiantes - 1; j++)
            {
                estudiantes[j] = estudiantes[j + 1];
            }
            (*numEstudiantes)--;
            encontrado = 1;
        }
    }

    if (encontrado == 0)
    {
        printf("NO SE ENCONTRO LA MATRICULA\n");
    }
    else
    {
        printf("EL REGISTRO FUE ELIMINADO\n");
    }
}

void imprimir(struct InfoAlumno estudiantes[], int numEstudiantes)
{
    int i;
    if (numEstudiantes == 0)
    {
        printf("NO HAY REGISTROS PARA IMPRIMIR\n");
        return;
    }

    for (i = 0; i < numEstudiantes; i++)
    {
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        printf("Estudiante #%d\n", i + 1);
        printf("Nombre: %s %s %s\n", estudiantes[i].Nombre, estudiantes[i].ApPat,
estudiantes[i].ApMat);
        printf("Sexo: %s\n", estudiantes[i].sexo);
        printf("Matricula: %d\n", estudiantes[i].matricula);
        printf("Estado: %s\n", estudiantes[i].estado);
        printf("Edad: %d\n", estudiantes[i].edad);
        printf("\n");
    }
}

void buscar(struct InfoAlumno estudiantes[], int *numEstudiantes)
{
    int i, matricula, encontrado = 0;

    matricula = validarNumeros("INGRESE LA MATRICULA DEL ALUMNO PARA ENCONTRAR SU
REGISTRO: ", 100000, 500000);

    for (i = 0; i <= *numEstudiantes; i++)
    {
        if (estudiantes[i].matricula == matricula)
        {
            printf("Estudiante #%d\n", i + 1);
            printf("Nombre: %s %s %s\n", estudiantes[i].Nombre,
estudiantes[i].ApPat, estudiantes[i].ApMat);
            printf("Sexo: %s\n", estudiantes[i].sexo);
            printf("MatrIcula: %d\n", estudiantes[i].matricula);
            printf("Estado: %s\n", estudiantes[i].estado);
            printf("Edad: %d\n", estudiantes[i].edad);
            printf("\n");
            encontrado = 1;
        }
    }

    if (encontrado == 0)
    {
        printf("NO SE ENCONTRO LA MATRICULA\n");
    }
}

void ordenar(struct InfoAlumno estudiantes[], int numEstudiantes)
{
    int i, j;
    struct InfoAlumno temp;

    for (i = 0; i < numEstudiantes - 1; i++)
    {
        for (j = 0; j < numEstudiantes - i - 1; j++)
        {
            if (estudiantes[j].matricula > estudiantes[j + 1].matricula)
            {
                temp = estudiantes[j];
                estudiantes[j] = estudiantes[j + 1];
                estudiantes[j + 1] = temp;
            }
        }
    }
}
```



- Librería personal

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdbool.h>
#include <ctype.h>

int validarNumeros(char mensaje[], int ri, int rf);
int validarLetras(const char mensaje[], char nombre[], int tam);
int Buscar(int vect[], int n, int num);
int repetir();
int VectSinRep(int vec[], int n, int ri, int rf);
int random(int ri, int rf);
int imprimirVec(char msg[], int vec[], int n);
void matriz4x4(int matriz[4][4], int ri, int rf);
void imprimirMatriz(int matriz[][4], int m, int n, char mensaje[]);
void ordenarVec(int vec[], int n);

void buscarValor(int vec[], int n)
{
    int i, encontrado = 0, numero;
    numero = validarNumeros("INGRESE EL NUMERO: ", 0, 10000);
    for (i = 0; i < n; i++)
    {
        if (vec[i] == numero)
        {
            printf("El numero %d se encuentra en el lugar [%d]\n", numero, i);
            encontrado = 1;
        }
    }
    if (encontrado == 0)
    {
        printf("El numero no se encuentra en el vector\n");
    }
}

void ordenarVec(int vec[], int n)
{
    int i, j, temp;
    int ordenado = 0;

    for (i = 0; i < n - 1; i++)
    {
        ordenado = 1;
        for (j = 0; j < n - i - 1; j++)
        {
            if (vec[j] > vec[j + 1])
            {
                temp = vec[j];
                vec[j] = vec[j + 1];
                vec[j + 1] = temp;
                ordenado = 0;
            }
        }
    }
}
```



```
    }  
    }  
}  
  
void imprimirMatriz(int matriz[][4], int m, int n, char mensaje[])  
{  
    int i, j;  
    printf("%s:\n", mensaje);  
    for (i = 0; i < m; i++)  
    {  
        for (j = 0; j < n; j++)  
        {  
            printf("[%3d]  ", matriz[i][j]);  
        }  
        printf("\n");  
    }  
}  
  
void matriz4x4(int matriz[4][4], int ri, int rf)  
{  
    srand(time(NULL));  
    int i, j, repe, vec[16];  
    for (i = 0; i < 16; i++)  
    {  
        do  
        {  
            vec[i] = random(ri, rf);  
            repe = 0;  
            for (j = 0; j < i; j++)  
            {  
                if (vec[i] == vec[j])  
                {  
                    repe = 1;  
                }  
            }  
        } while (repe == 1);  
    }  
    for (i = 0; i < 4; i++)  
    {  
        for (j = 0; j < 4; j++)  
        {  
            matriz[i][j] = vec[i * 4 + j];  
        }  
    }  
}  
  
int validarNumeros(char mensaje[], int ri, int rf)  
{  
    char entrada[100];  
    int num;  
    do  
    {  
        printf("%s", mensaje);  
        fgets(entrada, sizeof(entrada), stdin);  
        num = atoi(entrada);  
        if (num < ri || num > rf)  
        {  
            printf("\nINGRESE EL NUMERO OTRA VEZ PORFAVOR\n");  
        }  
    } while (num < ri || num > rf);  
}
```




```
    return num;
}

int validarLetras(const char mensaje[], char nombre[], int tam)
{
    char entrada[1000];
    int i;
    int entradaValida;

    do
    {
        entradaValida = 1;
        printf("%s", mensaje);
        fgets(entrada, sizeof(entrada), stdin);
        entrada[strcspn(entrada, "\n")] = '\0';

        for (i = 0; entrada[i] != '\0'; i++)
        {
            if (!isalpha(entrada[i]))
            {
                entradaValida = 0;
                printf("Cadena no válida. Deben ser solo letras.\n");
                break;
            }
            else
            {
                entrada[i] = toupper(entrada[i]);
            }
        }
    } while (!entradaValida);

    strncpy(nombre, entrada, tam - 1);
    nombre[tam - 1] = '\0';
}

int repetir()
{
    int repetir;
    repetir = validarNumeros("QUIERES HACER OTRA OPCION\nSI=1\nNO=2\n", 1, 2);
    if (repetir == 1)
    {
        return 1;
    }
    else
    {
        printf("FIN DEL PROGRAMA");
        return 0;
    }
}

int Buscar(int vect[], int n, int num)
{
    int i;
    for (i = 0; i < n; i++)
    {
        if (vect[i] == num)
        {
            return i;
        }
    }
}
```



```
        return -1;
    }

    int VectSinRep(int vec[], int n, int ri, int rf)
    {
        int i, num;
        for (i = 0; i < n; i++)
        {
            do
            {
                num = random(ri, rf);
            } while (Buscar(vec, i, num) != -1);
            vec[i] = num;
        }
    }

    int random(int ri, int rf)
    {
        int r;
        r = (rf - ri) + 1;
        return (rand() % r) + ri;
    }

    int imprimirVec(char msg[], int vec[], int n)
    {
        int i;
        printf("%s\n", msg);
        for (i = 0; i < n; i++)
        {
            printf("[%1d]\n", vec[i]);
        }
    }
}
```

- Resultados



```
===MENU===
1.- AGREGAR
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO (logico)
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0.- SALIR
1
QUIERES HACER OTRA OPCION
SI=1
NO=2
1

===MENU===
1.- AGREGAR
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO (logico)
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0.- SALIR
2
TIENES 1 O 2 NOMBRES?
2
INGRESE EL PRIMER NOMBRE: daniela
INGRESE EL SEGUNDO NOMBRE: lizeth
INGRESE EL APELLIDO PATERNO: rodriguez
INGRESE EL APELLIDO MATERNO: retana
INGRESE LA MATRICULA: 372813
INGRESE LA EDAD: 19
INGRESE EL NUMERO DEL GÉNERO:
0.- HOMBRE
1.- MUJER
1
CUAL ES EL ESTADO ACADEMICO?
```



```
0.- DADO DE BAJA
1.- BAJA TEMPORAL
2.- MATRICULADO
3.- EGRESADO
2
QUIERES HACER OTRA OPCION
SI=1
NO=2
1

===MENU===
1.- AGREGAR
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO (logico)
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0.- SALIR
6
Estudiante #1
Nombre: FERNANDO ROMERO SANCHEZ
Sexo: HOMBRE
Matricula: 112712
Estado: MATRICULADO
Edad: 20

Estudiante #2
Nombre: ADAN ROJAS MORALES
Sexo: HOMBRE
Matricula: 123712
Estado: BAJA TEMPORAL
Edad: 26
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Estudiante #3

Nombre: PEDRO RAUL MENDOZA DIAZ

Sexo: HOMBRE

Matricula: 106296

Estado: BAJA TEMPORAL

Edad: 37

Estudiante #4

Nombre: MIGUEL ANDRES MEDINA DIAZ

Sexo: HOMBRE

Matricula: 113763

Estado: BAJA TEMPORAL

Edad: 24

Estudiante #5

Nombre: DAVID GABRIEL SANTOS SANCHEZ

Sexo: HOMBRE

Matricula: 112204

Estado: BAJA TEMPORAL

Edad: 40

Estudiante #6

Nombre: ROSA ROJAS FLORES

Sexo: MUJER

Matricula: 117494

Estado: DADO DE BAJA

Edad: 18

Estudiante #7

Nombre: ELENA RODRIGUEZ

Sexo: MUJER

Matricula: 127292

Estado: EGRESADO

Edad: 18



Estudiante #8

Nombre: MARTA PATRICIA MIRANDA RODRIGUEZ

Sexo: MUJER

Matricula: 103903

Estado: DADO DE BAJA

Edad: 18

Estudiante #9

Nombre: JORGE RIOS RUIZ

Sexo: HOMBRE

Matricula: 114789

Estado: EGRESADO

Edad: 23

Estudiante #10

Nombre: SERGIO ALEJANDRO GOMEZ GONZALEZ

Sexo: HOMBRE

Matricula: 120628

Estado: DADO DE BAJA

Edad: 32

Estudiante #11

Nombre: DANIELA LIZETH RODRIGUEZ RETANA

Sexo: MUJER

Matricula: 372813

Estado: MATRICULADO

Edad: 19

QUIERES HACER OTRA OPCION

SI=1

NO=2

1



===MENU===

- 1.- AGREGAR
- 2.- AGREGAR MANUAL
- 3.- ELIMINAR REGISTRO (logico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 0.- SALIR

5

QUIERES HACER OTRA OPCION

SI=1

NO=2

1

===MENU===

- 1.- AGREGAR
- 2.- AGREGAR MANUAL
- 3.- ELIMINAR REGISTRO (logico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 0.- SALIR

6

Estudiante #1

Nombre: MARTA PATRICIA MIRANDA RODRIGUEZ

Sexo: MUJER

Matricula: 103903

Estado: DADO DE BAJA

Edad: 18

Estudiante #2

Nombre: PEDRO RAUL MENDOZA DIAZ

Sexo: HOMBRE

Matricula: 106296

Estado: BAJA TEMPORAL

Edad: 37



Estudiante #3

Nombre: DAVID GABRIEL SANTOS SANCHEZ

Sexo: HOMBRE

Matricula: 112204

Estado: BAJA TEMPORAL

Edad: 40

Estudiante #4

Nombre: FERNANDO ROMERO SANCHEZ

Sexo: HOMBRE

Matricula: 112712

Estado: MATRICULADO

Edad: 20

Estudiante #5

Nombre: MIGUEL ANDRES MEDINA DIAZ

Sexo: HOMBRE

Matricula: 113763

Estado: BAJA TEMPORAL

Edad: 24

Estudiante #6

Nombre: JORGE RIOS RUIZ

Sexo: HOMBRE

Matricula: 114789

Estado: EGRESADO

Edad: 23

Estudiante #7

Nombre: ROSA ROJAS FLORES

Sexo: MUJER

Matricula: 117494

Estado: DADO DE BAJA

Edad: 18



Estudiante #8

Nombre: SERGIO ALEJANDRO GOMEZ GONZALEZ

Sexo: HOMBRE

Matricula: 120628

Estado: DADO DE BAJA

Edad: 32

Estudiante #9

Nombre: ADAN ROJAS MORALES

Sexo: HOMBRE

Matricula: 123712

Estado: BAJA TEMPORAL

Edad: 26

Estudiante #10

Nombre: ELENA RODRIGUEZ

Sexo: MUJER

Matricula: 127292

Estado: EGRESADO

Edad: 18

Estudiante #11

Nombre: DANIELA LIZETH RODRIGUEZ RETANA

Sexo: MUJER

Matricula: 372813

Estado: MATRICULADO

Edad: 19

QUIERES HACER OTRA OPCION

SI=1

NO=2

1



```
===MENU===
1.- AGREGAR
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO (logico)
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0.- SALIR
4
INGRESE LA MATRICULA DEL ALUMNO PARA ENCONTRAR SU REGISTRO: 372813
Estudiante #11
Nombre: DANIELA LIZETH RODRIGUEZ RETANA
Sexo: MUJER
Matricula: 372813
Estado: MATRICULADO
Edad: 19

QUIERES HACER OTRA OPCION
SI=1
NO=2
1

===MENU===
1.- AGREGAR
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO (logico)
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0.- SALIR
3
INGRESE LA MATRICULA DEL ALUMNO PARA ELIMINAR SU REGISTRO: 372813
EL REGISTRO FUE ELIMINADO
QUIERES HACER OTRA OPCION
SI=1
NO=2
1
```



===MENU===

- 1.- AGREGAR
- 2.- AGREGAR MANUAL
- 3.- ELIMINAR REGISTRO (logico)
- 4.- BUSCAR
- 5.- ORDENAR
- 6.- IMPRIMIR
- 0.- SALIR

6

Estudiante #1

Nombre: MARTA PATRICIA MIRANDA RODRIGUEZ

Sexo: MUJER

Matricula: 103903

Estado: DADO DE BAJA

Edad: 18

Estudiante #2

Nombre: PEDRO RAUL MENDOZA DIAZ

Sexo: HOMBRE

Matricula: 106296

Estado: BAJA TEMPORAL

Edad: 37

Estudiante #3

Nombre: DAVID GABRIEL SANTOS SANCHEZ

Sexo: HOMBRE

Matricula: 112204

Estado: BAJA TEMPORAL

Edad: 40

Estudiante #4

Nombre: FERNANDO ROMERO SANCHEZ

Sexo: HOMBRE

Matricula: 112712

Estado: MATRICULADO

Edad: 20



Estudiante #5

Nombre: MIGUEL ANDRES MEDINA DIAZ

Sexo: HOMBRE

Matricula: 113763

Estado: BAJA TEMPORAL

Edad: 24

Estudiante #6

Nombre: JORGE RIOS RUIZ

Sexo: HOMBRE

Matricula: 114789

Estado: EGRESADO

Edad: 23

Estudiante #7

Nombre: ROSA ROJAS FLORES

Sexo: MUJER

Matricula: 117494

Estado: DADO DE BAJA

Edad: 18

Estudiante #8

Nombre: SERGIO ALEJANDRO GOMEZ GONZALEZ

Sexo: HOMBRE

Matricula: 120628

Estado: DADO DE BAJA

Edad: 32

Estudiante #9

Nombre: ADAN ROJAS MORALES

Sexo: HOMBRE

Matricula: 123712

Estado: BAJA TEMPORAL

Edad: 26



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Estudiante #10

Nombre: ELENA RODRIGUEZ

Sexo: MUJER

Matricula: 127292

Estado: EGRESADO

Edad: 18

QUIERES HACER OTRA OPCION

SI=1

NO=2

1

===MENU===

1.- AGREGAR

2.- AGREGAR MANUAL

3.- ELIMINAR REGISTRO (logico)

4.- BUSCAR

5.- ORDENAR

6.- IMPRIMIR

0.- SALIR

4

INGRESE LA MATRICULA DEL ALUMNO PARA ENCONTRAR SU REGISTRO: 272813

NO SE ENCONTRO LA MATRICULA

QUIERES HACER OTRA OPCION

SI=1

NO=2

1

===MENU===

1.- AGREGAR

2.- AGREGAR MANUAL

3.- ELIMINAR REGISTRO (logico)

4.- BUSCAR

5.- ORDENAR

6.- IMPRIMIR

0.- SALIR

0

FIN DEL PROGRAMA!