

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



Ingeniería en computación

Actividad 13

Materia: Programación estructurada

ALUMNO: Villalobos Ensaldo Luis Daniel

MATRÍCULA: 368617

GRUPO: 441

PROFESOR: Pedro Nunez Yepiz

Ensenada, Baja California a 20 de mayo de 2024.

M E N U

- 1.- Agregar
- 2.- Editar registro
- 3.- Eliminar Registro
- 4.- Buscar
- 5.- Ordenar
- 6.- Imprimir
- 7.- Generar Archivo
- 8.- Ver archivo
- 9.- Generar binario
- 10.- Cargar binario
- 11.- Mostrar eliminados

Actividad13

- ≡ datos.dll
- 🕒 datos.tmp
- ≡ hola.txt
- 🔗 LDVE_PE_AC... 9+
- ≡ LDVE_PE_ACT13.exe
- 🌐 luis.h 4

94.-	331840	ANGEL	PEREZ	GARCIA	27	HOMBRE
No.	Matricula	Nombre	Apellidos		Edad	Sexo
95.-	332080	JUAN	SOTO	GARCIA	29	HOMBRE
No.	Matricula	Nombre	Apellidos		Edad	Sexo
96.-	332159	HANNA	DIAZ	GARCIA	18	MUJER
No.	Matricula	Nombre	Apellidos		Edad	Sexo
97.-	332362	ANGELA	IBARRA	ORTIZ	18	MUJER
No.	Matricula	Nombre	Apellidos		Edad	Sexo
98.-	332463	ISRAEL	IBARRA	OROZCO	23	HOMBRE
No.	Matricula	Nombre	Apellidos		Edad	Sexo
99.-	332565	JUAN	HERNANDEZ	CHAVEZ	26	HOMBRE

Matricula que busca: 332080

La matricula esta en: 95

No.	Matricula	Nombre	Apellidos		Edad	Sexo
95.-	332080	JUAN	SOTO	GARCIA	29	HOMBRE

Presione una tecla para continuar . . .

Que quiere editar?

- (1)Matricula
- (2)Nombre
- (3)Apellido
- (4)2do Apellido
- (5)Edad
- (6)Sexo
- (7)Eliminar
- (0)Cancelar

Escoja opcion: █

Escoja opcion: 1

MATRICULA

Nueva matricula: 332080

Matricula ya usada

Nueva matricula: █

Archivo a leer: hola█

2483.-	332491	JACOB	VICTORIO	CASAS	25	HOMBRE
2484.-	332511	JUAN	DOMINGUEZ	CARMINE	24	HOMBRE
2485.-	332520	ANA	VICTORIO	CASAS	18	MUJER
2486.-	332552	ANGEL	DOMINGUEZ	CASAS	19	HOMBRE
2487.-	332569	PAULA	ESPINOZA	CARMINE	26	MUJER
2488.-	332581	MARIA	DIAZ	OROZCO	26	MUJER
2489.-	332595	LUIS	ESPINOZA	CARMINE	25	HOMBRE
2490.-	332599	VIVIANA	HERNANDEZ	WAYNE	20	MUJER
2491.-	332621	ANGEL	HERNANDEZ	ORTIZ	21	HOMBRE
2492.-	332633	LUIS	HERNANDEZ	CRUZ	28	HOMBRE
2493.-	332650	LOLA	IBARRA	PAREDES	23	MUJER
2494.-	332662	HANNA	VICTORIO	OROZCO	22	MUJER
2495.-	332712	ALAN	VILLALOBOS	ORTIZ	23	HOMBRE
2496.-	332714	ANGELA	VICTORIO	PAREDES	19	MUJER
2497.-	332730	PAOLA	DIAZ	RUIZ	26	MUJER
2498.-	332737	GABRIEL	VICTORIO	ACOSTA	18	HOMBRE
2499.-	332745	ALAN	VICTORIO	TORRES	24	HOMBRE

Presione una tecla para continuar . . . █

PROBLEMS 28 OUTPUT TERMINAL ... [?] LDVE_PE_ACT13

No hay registros eliminados

Presione una tecla para continuar . . . █

```

Matricula que busca: 300366
La matricula esta en: 292
No.   Matricula  Nombre  Apellidos          Edad  Sexo
292.- 300366    MARIA   LEAL      PAREDES           23   MUJER
(0)Cancelar
(1)Eliminar
Seleccione: 1

```

```

No.   Matricula  Nombre  Apellidos          Edad  Sexo
193.- 331272    SONIA  DOMINGUEZ    CHAVEZ            27   MUJER
No.   Matricula  Nombre  Apellidos          Edad  Sexo
194.- 331293    WALTER  SOTO        CRUZ              24   HOMBRE
No.   Matricula  Nombre  Apellidos          Edad  Sexo
195.- 331778    ISRAEL  VICTORIO    WAYNE             21   HOMBRE
No.   Matricula  Nombre  Apellidos          Edad  Sexo
292.- 300366    MARIA   LEAL      PAREDES           23   MUJER
Presione una tecla para continuar . . .

```

Programa1

```

Talum Generar_reg(Talum registros[], int m)
{
    Talum reg;
    reg.status = 1;
    char nombres[20][20] = {"JUAN", "LUIS", "DANIEL", "SAUL", "JACOB",
"ANGEL", "ALAN", "YAHIR", "GABRIEL", "ISRAEL", "WALTER"};
    char mujeres[20][20] = {"ANGELA", "SOFIA", "VIVIANA", "ANA",
"MARIA", "PAOLA", "PAULA", "SONIA", "VANESSA", "HANNA", "LOLA"};
    char appat[20][20] = {"SOTO", "VILLALOBOS", "VICTORIO", "PEREZ",
"PINEDA", "HERNANDEZ", "LEAL", "ESPINOZA", "IBARRA", "DIAZ",
"DOMINGUEZ", "MARTINEZ"};
    char apmat[20][20] = {"WAYNE", "CRUZ", "CHAVEZ", "CARMINE", "RUIZ",
"ORTIZ", "TORRES", "OROZCO", "GARCIA", "PAREDES", "ACOSTA", "CASAS"};
    int sexo;
    reg.edad = (rand() % 12) + 18;

    strcpy(reg.appat, appat[rand() % 12]);
    strcpy(reg.apmat, apmat[rand() % 12]);
    sexo = (rand() % 2) + 1;
    if (sexo == 1)
    {
        strcpy(reg.sexo, "HOMBRE");
    }
}

```

```
        copystr(reg.nombre, nombres[rand() % 11]);
    }
    else
    {
        copystr(reg.sexo, "MUJER");
        copystr(reg.nombre, mujeres[rand() % 11]);
    }
    int resul;
    do
    {
        reg.matricula = (rand() % 99999) + 300000;
        resul = Buscar(registros, m, reg.matricula);
    } while (resul != -1);

    return reg;
}
```

```

int Buscar(Talum registros[], int m, int matri, short ordenado)
{
    int i;
    if (ordenado == 1)
    {
        int bajo = 0, centro, alto = m - 1;
        do
        {
            centro = (bajo + alto) / 2;
            if (registros[centro].matricula < matri)
            {
                bajo = centro + 1;
            }
            else
            {
                if (registros[centro].matricula == matri)
                {
                    return centro;
                }
                else
                {
                    alto = centro - 1;
                }
            }
        } while (alto >= bajo);
    }
    else
    {
        for (i = 0; i < m; i++)
        {
            if (registros[i].matricula == matri)
            {
                return i;
            }
        }
    }
    return -1;
}

short Editar_reg(Talum registros[], int m, short ordenado)
{
    system("CLS");
    int pos, op = -1;
    pos = Case_4(registros, m, ordenado);
}

```

```

        if (pos != -1 && registros[pos].status != 0)
        {
            do
            {
                printf("Que quiere editar?\n");
                printf("(1)Matricula\n(2)Nombre\n(3)Apellido\n(4)2do
Apellido\n");
                printf("(5)Edad\n(6)Sexo\n(7)Eliminar\n(0)Cancelar\n");
                op = valinum(0, 7, "Escoja opcion: ");
                ordenado = Ediciones(registros, m, pos, op, ordenado);
                printf("Registro actualizado:\n");
                Print_reg(registros[pos], pos);
            } while (op != 0);
            system("PAUSE");
        }
        else
        {
            if (registros[pos].status == 0)
            {
                printf("Solo se pueden editar registros activos\n");
                system("PAUSE");
            }
        }
        return ordenado;
    }
}

short Ediciones(Talum registros[], int m, int pos, int op, short
ordenado)
{
    char nombre[20];
    switch (op)
    {
        case 1:
            long matri;
            short resul;
            printf("          MATRICULA\n");
            do
            {
                matri = valilong(300000, 399999, "Nueva matricula: ");
                // validar que no se repita la matricula
                resul = Buscar(registros, m, matri, ordenado);
                if (resul != -1)
                {
                    printf("Matricula ya usada\n");
                }
            }
        }
    }
}

```

```

    }
    } while (resul != -1);
    registros[pos].matricula = matri;
    ordenado = 2;
    break;
case 2:
    printf("          NOMBRE\n");
    strscan(nombre, "Ingrese nuevo nombre: ");
    valichar(nombre);
    copystr(registros[pos].nombre, nombre);
    break;
case 3:
    printf("          APELLIDO\n");
    strscan(nombre, "Ingrese nuevo nombre: ");
    valichar(nombre);
    copystr(registros[pos].appat, nombre);
    break;
case 4:
    printf("          2do APELLIDO\n");
    strscan(nombre, "Ingrese nuevo nombre: ");
    valichar(nombre);
    copystr(registros[pos].apmat, nombre);
    break;
case 5:
    int edad;
    printf("          EDAD\n");
    edad = valinum(1, 90, "Ingrese nueva edad: ");
    registros[pos].edad = edad;
    break;
case 6:
    bool sex;
    printf("          SEXO\n");
    sex = valinum(0, 1, "(1)Hombre\n(0)Mujer\nEscoja sexo: ");
    if (sex)
    {
        copystr(registros[pos].sexo, "HOMBRE");
    }
    else
    {
        copystr(registros[pos].sexo, "MUJER");
    }
    break;
case 7:

```



```

        bool eliminar;
        printf("                ELIMINAR\n");
        eliminar = valinum(0, 1, "(0)Cancelar\n(1)Eliminar\nSeleccione:
");
        if (eliminar)
        {
            registros[pos].status = 0;
            printf("Se elimino el registro\n");
        }
        else
        {
            printf("Cancelando\n");
        }
        break;
    }
    return ordenado;
}

int Case_4(Talum registros[], int m, short ordenado)
{
    system("CLS");
    int num, resul;
    num = valilong(300000, 399999, "Matricula que busca: ");
    resul = Buscar(registros, m, num, ordenado);
    if (resul == -1)
    {
        printf("No se encontro la matricula\n");
    }
    else
    {
        printf("La matricula esta en: %d\n", (resul));
        Print_reg(registros[resul], resul);
    }
    system("PAUSE");
    return resul;
}

void Eliminar(Talum registros[], int m, short ordenado)
{
    long num;
    int resul;
    bool eliminar;

    num = valilong(300000, 399999, "Matricula que busca: ");
    resul = Buscar(registros, m, num, ordenado);

```

```

        if (resul == -1)
        {
            printf("No se encontro la matricula\n");
        }
        else
        {
            printf("La matricula esta en: %d\n", (resul));
            Print_reg(registros[resul], resul);
            eliminar = valinum(0, 1, "(0)Cancelar\n(1)Eliminar\nSeleccione:");
        }

        if (eliminar)
        {
            registros[resul].status = 0;
            printf("Se elimino el registro\n");
        }
        else
        {
            printf("Cancelando\n");
        }
    }
}

int Ordenar(Talum registros[], int m, short ordenado)
{
    int j;
    Talum temp;
    if (ordenado)
    {
        if (ordenado == 2) // mas o menos desordenado
        {
            MergeSort(registros, 0, m - 1);
        }
        else
        {
            for (int i = 0; i < m - 1; i++)
            {
                for (j = i + 1; j < m; j++)
                {
                    if (registros[j].matricula <
registros[i].matricula)
                    {
                        temp = registros[i];
                        registros[i] = registros[j];
                        registros[j] = temp;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}

else
{
    QuickSort(registros, 0, m - 1);
}

return 1;
}

void Mostrar_todo(Talum registros[], int m)
{
    for (int i = 0; i < m; i++)
    {
        if (registros[i].status == 1)
        {
            Print_reg(registros[i], i);
        }
    }
}

void Print_reg(Talum registros, int pos)
{
    printf("No.    Matricula  Nombre    Apellidos          Edad\n");
    printf("%3d.- %6ld %10s %10s %10s %8d %8s\n", pos,
registros.matricula, registros.nombre, registros.appat,
registros.apmat, registros.edad, registros.sexo);
}

```

Programa7

```
void Gen_Arch(Talum registros[], int m)
{
    char nombre[20];
    system("CLS");
    strscan(nombre, "Escriba el nombre del archivo: ");
    /* printf("Escriba el nombre del archivo: ");
    fflush(stdin);
    gets(nombre); */
    valichar(nombre, "Intenta otro nombre: ");
    // concatenar nombre con txt
    strcat(nombre, ".txt");
    FILE *archivo = fopen(nombre, "w");
    if (archivo)
    {
        // Escribir en el archivo
        for (int i = 0; i < m; i++)
        {
            if (registros[i].status == 1)
            {
                fprintf(archivo, "%3d.- %6ld%10s%10s%10s%3d%8s\n", (i),
registros[i].matricula, registros[i].nombre, registros[i].appat,
registros[i].apmat, registros[i].edad, registros[i].sexo);
            }
        }
        printf("Se ha escrito el archivo\n");
    }
    fclose(archivo);
    printf("Se ha escrito el archivo\n");
    system("PAUSE");
}
```

Programa8

```
void Printtxt(void)
{
    system("CLS");
    int caracter;
    char nombre[20];
    strscan(nombre, "Archivo a leer: ");
    valichar(nombre);
    strcat(nombre, ".txt");
    FILE *arch;
    arch = fopen(nombre, "r");
    if (arch)
```

```

{
    while ((character = fgetc(arch)) != EOF)
    {
        putchar(character);
    }
}
else
{
    printf("No se pudo abrir el archivo\n");
}
system("PAUSE");
}

```

Programa9

```

void Crear_bin(Talum registros[], int m)
{
    system("CLS");
    Talum reg;
    FILE *fa;
    FILE *arch;
    fa = fopen("datos.dll", "r+b");
    arch = fopen("datos.tmp", "w+b");
    if (fa)
    {
        // leer
        while (fread(&reg, sizeof(Talum), 1, fa))
        {
            fwrite(&reg, sizeof(Talum), 1, arch);
        }
        fclose(fa);
        fclose(arch);
        printf("Se ha respaldado el archivo\n");
    }
    else
    {
        printf("No se puede respaldar el archivo\n");
    }
    // sobrescribir
    fa = fopen("datos.dll", "w+b");
    if (fa)
    {
        for (int i = 0; i < m; i++)
        {
            reg = registros[i];

```

```

        fwrite(&reg, sizeof(Talum), 1, fa);
    }
    fclose(fa);
    printf("Se ha escrito el archivo\n");
}
system("PAUSE");
}

```

Programa10

```

int Cargar_bin(Talum vect[], int m, int reg_act)
{
    system("CLS");
    Talum reg;
    FILE *fa;
    int i = reg_act;
    if (valinum(0, 1, "1 tmp 0 dll\n"))
    {
        fa = fopen("datos.tmp", "r+b");
    }
    else
    {
        fa = fopen("datos.dll", "r+b");
    }
    if (fa)
    {
        while (fread(&reg, sizeof(Talum), 1, fa) && i < m)
        {
            vect[i++] = reg;
        }
        fclose(fa);
        printf("Se ha cargado el archivo\n");
    }
    else
    {
        printf("No se pudo abrir el archivo\n");
    }
    system("PAUSE");
    return i;
}

```

Programa11

```

void Eliminados(void)
{
    system("CLS");
    int i = 0;
}

```

```

bool elim = 1;
Talum reg;
FILE *fa;
fa = fopen("datos.dll", "r+b");
if (fa)
{
    while (fread(&reg, sizeof(Talum), 1, fa))
    {
        if (reg.status == 0)
        {
            Print_reg(reg, i);
            elim = 0;
        }
        i++;
    }
    fclose(fa);
    if (elim)
    {
        printf("No hay registros eliminados\n");
    }
}
else
{
    printf("No se puede abrir el archivo\n");
}
system("PAUSE");
}

```