



**Universidad Autónoma de Baja California**  
**Facultad de Ingeniería Arquitectura y Diseño**



Ingeniero en Software y Tecnologías Emergentes.

Programación estructurada

**FUNCIONES y METODOS DE ORDENACION Y BUSQUEDA**  
**ESTRUCTURAS Y LIBRERIAS (p2)**

Actividad 11

Danna Guadalupe Sandez Islas

373080

# MENÚ

código:

```
do
{
    system("CLS");
    printf(" ---- MENU ----\n");
    printf("1- Agregar (manual)\n");
    printf("2- Agregar (automatico)\n");
    printf("3- Eliminar registro\n");
    printf("4- Buscar\n");
    printf("5- Ordenar\n");
    printf("6- Imprimir\n");
    printf("7- Archivo de texto\n");
    printf("0- SALIR\n");
    op = validar("Ingresa una opcion NUMERICA: ", 0, 7);
    system("CLS");
}
```

Salida:

```
---- MENU ----
1- Agregar (manual)
2- Agregar (automatico)
3- Eliminar registro
4- Buscar
5- Ordenar
6- Imprimir
7- Archivo de texto
0- SALIR
Ingresa una opcion NUMERICA: █
```

## 1.- Agregar

Código:

```
switch (op)
{
case 1: // agregar 1 registro manualmente
    if (k < 2000)
    {
        persona = manual();
        while (busq_seq(registro, i, persona.matricula) != -1) // validar la matricula
        {
            persona.matricula = validar("Por favor ingresa otra matricula: ", 30000, 399999);
        }
        registro[i++] = persona; // agregar la variable tipo Tdatos al registro
        k++; // contador para numero de registros creados
        printf("Registro lleno con éxito!\n");
    }
    else
    {
        printf("El registro llego a su maxima capacidad\n");
    }
    ordenado = 0; // bandera para saber si el registro esta ordenado
    system("PAUSE");
    break;
case 2: // agregar automatico
    if (k < 2000)
    {
        for (int j = 0; j < 100; j++)
        {
            persona = autom();
            while (busq_seq(registro, i, persona.matricula) != -1) // evaluar si la matricula se repite
            {
                persona.matricula = numAleatorio(30000, 399999);
            }
            registro[i++] = persona;
            k++;
        }
        printf("Registro lleno exitosamente con 100 personas!\n");
    }
    else
    {
        printf("El registro llego a su maxima capacidad\n");
    }
    ordenado = 0;
    system("PAUSE");
    break;
}
```

## 2.- Eliminar Registro

Código:

```
break;
case 3: // eliminar
    mat = validar("Ingrese matricula para eliminar: ", 300000, 399999);
    system("CLS");
    if (ordenado == 1)
    {
        encontrado = busq_binaria(registro, 300000, 399999, mat);
        if (encontrado != -1)
        {
            printf("MATRICULA: %d\nNOMBRE: %s\nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nFECHA DE NACIMIENTO: %02d/%02d/%d\nEDAD: %d\nSEXO: %s\nLUGAR DE NACIMIENTO: %s\nCURP: %s", registro[encontrado].matricula, registro[encontrado].nombre, registro[encontrado].apellido_paterno, registro[encontrado].apellido_materno, registro[encontrado].fecha_nacimiento, registro[encontrado].edad, registro[encontrado].sexo, registro[encontrado].lugar_nacimiento, registro[encontrado].curp);
            printf("\n\nDeseas eliminar al usuario %d?", registro[encontrado].matricula);
            op2 = validar("\n1-SI\n2-No\n", 1, 2);
            system("CLS");
            if (op2 == 1)
            {
                registro[encontrado].status = 0; // desactivar status
                printf("La matricula ahora es inactiva\n");
            }
        }
        else
        {
            printf("La matricula no se encuentra en el registro\n");
        }
    }
    else
    {
        encontrado = busq_seq(registro, i, mat);
        if (encontrado != -1) // si se encuentra la matricula en los registros
        {
            printf("MATRICULA: %d\nNOMBRE: %s\nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nFECHA DE NACIMIENTO: %02d/%02d/%d\nEDAD: %d\nSEXO: %s\nLUGAR DE NACIMIENTO: %s\nCURP: %s", registro[encontrado].matricula, registro[encontrado].nombre, registro[encontrado].apellido_paterno, registro[encontrado].apellido_materno, registro[encontrado].fecha_nacimiento, registro[encontrado].edad, registro[encontrado].sexo, registro[encontrado].lugar_nacimiento, registro[encontrado].curp);
            printf("\n\nDeseas eliminar al usuario %d?", registro[encontrado].matricula);
            op2 = validar("\n1-SI\n2-No\n", 1, 2);
            system("CLS");
            if (op2 == 1)
            {
                registro[encontrado].status = 0; // desactivar status
                printf("La matricula ahora es inactiva\n");
            }
        }
        else
        {
            printf("La matricula no se encuentra en el registro\n");
        }
    }
}
system("PAUSE");
break;
```

## 3.- Buscar

Código:

```
break;
case 4: // buscar
    mat = validar("Ingrese una matricula para buscar: ", 300000, 399999);
    system("CLS");
    if (ordenado == 1)
    {
        encontrado = busq_binaria(registro, 300000, 399999, mat);
        if (encontrado != -1) // si se encuentra
        {
            if (registro[encontrado].status == 0)
            {
                printf("Matricula inactiva\n");
            }
            else
            {
                printf("MATRICULA: %d\nNOMBRE: %s\nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nFECHA DE NACIMIENTO: %02d/%02d/%d\nEDAD: %d\nSEXO: %s\nLUGAR DE NACIMIENTO: %s\nCURP: %s\n", registro[encontrado].matricula, registro[encontrado].nombre, registro[encontrado].apellido_paterno, registro[encontrado].apellido_materno, registro[encontrado].fecha_nacimiento, registro[encontrado].edad, registro[encontrado].sexo, registro[encontrado].lugar_nacimiento, registro[encontrado].curp);
            }
        }
        else // no se encuentra
        {
            printf("La matricula no se encuentra en el registro\n");
        }
    }
    else
    {
        encontrado = busq_seq(registro, i, mat);
        if (encontrado != -1) // si se encuentra la matricula en los registros
        {
            if (registro[encontrado].status == 0) // alumno con status inactivo. NO HAY MATRICULA
            {
                printf("Matricula inactiva\n");
            }
            else // alumno activo, SI hay matricula
            {
                printf("MATRICULA: %d\nNOMBRE: %s\nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nFECHA DE NACIMIENTO: %02d/%02d/%d\nEDAD: %d\nSEXO: %s\nLUGAR DE NACIMIENTO: %s\nCURP: %s", registro[encontrado].matricula, registro[encontrado].nombre, registro[encontrado].apellido_paterno, registro[encontrado].apellido_materno, registro[encontrado].fecha_nacimiento, registro[encontrado].edad, registro[encontrado].sexo, registro[encontrado].lugar_nacimiento, registro[encontrado].curp);
            }
        }
        else // no se encuentra la matricula
        {
            printf("La matricula que buscas no se encuentra en el registro\n");
        }
    }
}
system("PAUSE");
break;
```

## 4.- Ordenar

Código:

```

        break;
    case 5: // ordenar
        if (ordenado == 1)
        {
            printf("Registro ya se encuentra ordenado\n");
        }
        else
        {
            if (i <= 500)
            {
                ordenado = ordenar(registro, i); // ordenas registro
            }
            else
            {
                ordenado = quicksort(registro, 0, i - 1);
            }
            printf("Registro ordenado\n");
        }
        system("PAUSE");
        break;

```

## 5.- Imprimir

Código:

```

        break;
    case 6: // imprimir
        print_reg(registro, i);
        break;

```

Función print\_reg:

```

void print_reg(Datos registro[], int j)
{
    int i, k, l = 0;
    printf("No. MATRICULA NOMBRE(S) APELLIDO PATERNO APELLIDO MATERNO FECHA DE NACIMIENTO EDAD SEXO LUGAR NAC CUR\n");
    for (i = 0; i < j; i++)
    {
        k = 0;
        while (k < 40)
        {
            if (registro[i].status == 1)
            {
                printf("%-18d %-20d %-23s %-29s %-26s %02d/%02d/%-26d %-18d %-15s %-23s %-5s\n", 1 + i, registro[i].matricula, registro[i].nom.nombre, registro[i].nom.ap_paterno, registro[i].nom.ap_materno, registro[i].nacimiento.dia,
                l++, k++);
            }
        }
        i += 40;
        system("PAUSE");
    }
}

```

## 6.- Archivo Texto

Código:

```

    case 7: // archivo de texto
        arch(registro, i);
        system("PAUSE");
        break;
    }
while (op != 0);

```

Función arch:

```

void arch(Datos registro[], int n)
{
    int i;
    FILE *fa;
    fa = fopen("C:\\Users\\User\\Desktop\\io.o\\Ser SEM\\PE\\Danna_Sandez_Programacion_Estructurada\\Danna_Sandez_Programacion_Estructurada\\Actividad_11\\output\\registros.txt", "w");
    fprintf(fa, "No. MATRICULA NOMBRE(S) APELLIDO PATERNO APELLIDO MATERNO FECHA DE NACIMIENTO EDAD SEXO LUGAR NAC\n");
    for (i = 0; i < n; i++)
    {
        if (registro[i].status == 1)
        {
            fprintf(fa, "%-18d %-20d %-23s %-29s %-26s %02d/%02d/%-26d %-18d %-15s %-23s %-5s\n", 1 + i, registro[i].matricula, registro[i].nom.nombre, registro[i].nom.ap_paterno, registro[i].nom.ap_materno, registro[i].nacimiento.dia,
            l++, k++);
        }
    }
    printf("Archivo escrito con exito\n");
    fclose(fa);
}

```