

Universidad Autónoma de Baja California



Ingeniería en Software y Tecnologías Emergentes.

**Actividad 10. Funciones Y Metodos De
Ordenacion Y Busqueda Estructuras Y Librerías**

Materia: Programación Estructurada.

Maestro: Pedro Yepiz

Alumno: Diego Quiros 372688

Fecha: 10/01/2023

INSTRUCCIONES:

- 1.- Realiza un programa en C que utilice una librería propia (Funciones de validar números y , cadenas)
- 2.- Realiza reporte de práctica
- 3.- Sube a Blackboard, programa, librería, y reporte de práctica y PDF anexo con capturas y código

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

Ejercicio 1. AGREGAR (AUTOM 10 REGISTROS).

```
C/C++
void fillAtmreg(Talumn vect_alum[], int indexNumber)
{
    int i, id;
    // generar matricula y status
    vect_alum[indexNumber].status = 1;

    do
    {
        id = 300000 + rand() % (399999 - 300000 + 1);
        i = searchLinearId(vect_alum, indexNumber, id);
    } while (i != -1);

    vect_alum[indexNumber].matricula = id;
    // Generar Nombre y apellidos
    genLastName(vect_alum[indexNumber].apPat);
    genLastName(vect_alum[indexNumber].apMat);
    genName(vect_alum[indexNumber].nombre);
    // Generar Otros datos
    vect_alum[indexNumber].edad = 17 + rand() % (40 - 17 + 1);
    vect_alum[indexNumber].sexo = sexDetect(vect_alum[indexNumber].nombre);
    indexNumber++;
}
```

```

//Generadores de nombres
void genName(char name[50])
{
    char maleName[25][35] = {
        "Alejandro", "Juan", "Manuel", "Jose", "Luis", "Carlos", "Pedro",
        "Antonio", "Javier", "Miguel", "David", "Francisco",
        "Raul", "Daniel", "Guillermo", "Alberto", "Mario", "Eduardo",
        "Angel", "Sergio", "Oscar", "Ruben", "Fernando", "Jorge", "Diego"};

    char femName[25][35] = {
        "Maria", "Ana", "Laura", "Isabel", "Carmen", "Rosa", "Patricia",
        "Sara", "Elena", "Andrea", "Marta", "Sofia",
        "Beatriz", "Natalia", "Clara", "Silvia", "Lucia", "Catalina",
        "Diana", "Lorena", "Vanessa", "Alicia", "Carolina", "Julieta"};

    int OneOrTwo = rand() % 2 + 1; // 1 para un solo nombre, 2 para dos
    int ForM = rand() % 2 + 1;     // 1 para masculino, 2 para femenino
    int namerand, namerand2;

    if (OneOrTwo == 1)
    {
        if (ForM == 1)
        {
            namerand = rand() % 25;
            strcpy(name, maleName[namerand]);
        }
        else if (ForM == 2)
        {
            namerand = rand() % 25;
            strcpy(name, femName[namerand]);
        }
    }
    else if (OneOrTwo == 2)
    {
        if (ForM == 1)
        {
            namerand = rand() % 25;
            // Comprobar que el segundo nombre no sea igual al primero
            do
            {
                namerand2 = rand() % 25;
            } while (namerand2 == namerand);

            sprintf(name, "%s %s", maleName[namerand], maleName[namerand2]);
        }
        else if (ForM == 2)
        {

```

```

        namerand = rand() % 25;
        // Comprobar que el segundo nombre no sea igual al primero
        do
        {
            namerand2 = rand() % 25;
        } while (namerand2 == namerand);

        sprintf(name, "%s %s", femName[namerand], femName[namerand2]);
    }
}

//Generador de apellidos
// Funcion que tiene una lista de apellidos y te da uno aleatorio
void genLastName(char lastname[30])
{
    char LastNamelist[30][25] = {"Gonzalez", "Rodriguez", "Lopez", "Perez",
    "Martinez", "Garcia", "Sanchez",
                                "Romero", "Fernandez", "Torres", "Ruiz",
    "Diaz",
                                "Jimenez", "Vargas", "Castro", "Morales",
    "Rivera", "Ortega", "Parra", "Yepiz", "Aguilar", "Navarro", "Ramirez",
    "Vicencio", "Llamas", "Diarte"};

    int namerand = rand() % 25;

    strcpy(lastname, LastNamelist[namerand]);
}

/*
    Funcion que detecta el sexo de el alumno mediante su nombre
    retorna 2 si es mujer y 1 si es hombre
*/
int sexDetect(char name[30])
{
    char femName[25][25] = {
        "Maria", "Ana", "Laura", "Isabel", "Carmen", "Rosa", "Patricia",
        "Sara", "Elena", "Andrea", "Marta", "Sofia",
        "Beatriz", "Natalia", "Clara", "Silvia", "Lucia", "Catalina",
        "Diana", "Lorena", "Vanessa", "Alicia", "Carolina", "Julieta"};

    char nametemp[30]; // Se genera una cadena temporal
    strcpy(nametemp, name); // se copia el nombre a comparar en el temporal

    // Se recorta la cadena hasta antes del primer espacio " ", en casod de
    que sean
    // dos nombres
    strtok(nametemp, " ");

```

```

for (int i = 0; i < 25; i++)
{
    if (strcmp(nametemp, femName[i]) == 0)
    {
        return 2;
    }
}
return 1;
}

```

Ejercicio 2. Agregar Manual.

C/C++

```

// Funcion que permite al usuario ingresar los datos para llenar un registro
void fillRegManual(Talumn vect_alum[], int indexNumber)
{
    int correct = 0;
    vect_alum[indexNumber].status = 1;
    printf("Llenado Manual\n");
    int i;
    do
    {
        printf("Ingresa Matricula: ");

        vect_alum[indexNumber].matricula = validInt(300000, 399999, "Matricula
Fuera de Rango\n");
        i = searchLinearId(vect_alum, indexNumber,
vect_alum[indexNumber].matricula);
        if (i != -1)
        {
            printf("Matricula Ya Registrada\n");
        }

    } while (i != -1);

    printf("Ingresa Apellido Paterno: ");
    fflush(stdin);
    gets(vect_alum[indexNumber].apPat);
    do
    {
        correct = validSingleString(vect_alum[indexNumber].apPat);
        if (correct == 0)
        {
            printf("Error - Reingresa correctamente los datos\n");
        }
    }
}

```

```

        printf("Ingresa Apellido Paterno: ");
        fflush(stdin);
        gets(vect_alum[indexNumber].apPat);
    }
} while (correct == 0);
capitalizeWords(vect_alum[indexNumber].apPat);

printf("Ingresa Apellido Materno: ");
fflush(stdin);
gets(vect_alum[indexNumber].apMat);
do
{
    correct = validSingleString(vect_alum[indexNumber].apMat);
    if (correct == 0)
    {
        printf("Error - Reingresa correctamente los datos\n");
        printf("Ingresa Apellido Materno: ");
        fflush(stdin);
        gets(vect_alum[indexNumber].apMat);
    }
} while (correct == 0);
capitalizeWords(vect_alum[indexNumber].apMat);

printf("Ingresa Nombre: ");
fflush(stdin);
gets(vect_alum[indexNumber].nombre);
do
{
    correct = validString(vect_alum[indexNumber].nombre);
    if (correct == 0)
    {
        printf("Error - Reingresa correctamente los datos\n");
        printf("Ingresa Nombre: ");
        fflush(stdin);
        gets(vect_alum[indexNumber].nombre);
    }
} while (correct == 0);
capitalizeWords(vect_alum[indexNumber].nombre);

printf("Ingresa Edad: ");
vect_alum[indexNumber].edad = validInt(17, 99, "Edad invalida\n");
printf("Ingresa sexo 1] Hombre 2] Mujer: ");
vect_alum[indexNumber].sexo = validInt(1, 2, "Opcion invalida");
indexNumber++;
}

```

Ejercicio 3. Eliminar Registro (Lógico).

C/C++

```
case 3:
    printRegs(_alum, indexNumber);
    printf("Ingresa matricula a dar de baja: ");
    id = validInt(300000, 399999, "Matricula fuera de rango");
    num = searchLinearId(_alum, indexNumber, id);
    if (num != -1)
    {
        _alum[num].status = 0;
        printf("Alumno dado de baja.\n");
    }
    else
    {
        printf("Esa matricula no esta registrada\n");
    }
    break;
```

Ejercicio 4. Buscar.

C/C++

```
//Busqueda Lineal
int searchLinearId(Talumn vector[], int m, int val)
{
    int i;

    for (i = 0; i < m; i++)
    {
        if (vector[i].matricula == val)
        {
            return i;
        }
    }
    return -1;
}

//Busqueda Binaria
int binarySearchId(Talumn vect[], int val, int start, int end)
{
    int mid;

    while (start <= end)
    {
        mid = (start + end) / 2;
        if (vect[mid].matricula == val)
```

```

        {
            return mid;
        }
        else if (val < vect[mid].matricula)
        {
            end = mid - 1;
        }
        else
        {
            start = mid + 1;
        }
    }

    return -1;
}

```

case 4:

```

printf("Ingresa matricula a buscar: ");
id = validInt(300000, 399999, "Numero invalido\n");
if (sorted == 0)
{
    num = searchLinearId(_alum, indexNumber, id);
    if (num != -1)
    {
        printIndividualRegs(_alum, num);
    }
    else
    {
        printf("Esa matricula no esta registrada\n");
    }
}
else
{
    num = binarySearchId(_alum, id, 0, indexNumber);
    if (num != -1)
    {
        printIndividualRegs(_alum, num);
    }
    else
    {
        printf("Esa matricula no esta registrada\n");
    }
}

break;

```


Ejercicio 5. Ordenar.

```
C/C++
/*
Funcion que sirve para ordenar la lista desordenada
mediante el metodo de la burbuja mejorada
*/
void sortIds(Talumn vector[], int m)
{
    int i, j;
    int aux;

    for (i = 0; i < m - 1; i++)
    {
        for (j = i + 1; j < m; j++)
        {
            if (vector[i].matricula > vector[j].matricula)
            {
                aux = vector[i].matricula;
                vector[i].matricula = vector[j].matricula;
                vector[j].matricula = aux;
            }
        }
    }
}

case 5:
if (sorted == 0)
{
    sortIds(_alum, indexNumber);
    sorted = 1; // Bandera que indica si los registros estan ordenados o no
}
else
{
    printf("Registros Actualmente ordenados\n");
}
break;
```

Ejercicio 6. Imprimir.

```
C/C++
/*
Funcion que imprimira todos los registros existentes
cuyo status sea igual a 1
*/
void printRegs(Talumn vect_alumn[], int index)
{

```

```

int i;
char sexo[2][10] = {"Hombre", "Mujer"};
char sexualumn[10];
printf("Matricula      ApPat      ApMat      Nombre
Edad      Sexo\n");

for (i = 0; i < index; i++)
{
    if (vect_alumn[i].sexo == 1)
    {
        strcpy(sexalumn, sexo[0]);
    }
    else
    {
        strcpy(sexalumn, sexo[1]);
    }
    if (vect_alumn[i].status == 1)
    {
        printf("%-6d      %-15s      %-15s      %-25s%2d      %-10s\n",
vect_alumn[i].matricula, vect_alumn[i].apPat, vect_alumn[i].apMat,
vect_alumn[i].nombre, vect_alumn[i].edad, sexualumn);
    }
}
}

```

Ejecución del Programa.

```

R E G I S T R O S U A B C
1.- AGREGAR (AUTOM 10 REGISTROS)
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0. Salir
Ingresa funcion a utilizar:

```

```
R E G I S T R O S U A B C
1.- AGREGAR (AUTOM 10 REGISTROS)
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0. Salir
Ingresa funcion a utilizar:
1
Presione una tecla para continuar . . .
```

```
R E G I S T R O S U A B C
1.- AGREGAR (AUTOM 10 REGISTROS)
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0. Salir
Ingresa funcion a utilizar:
2
Llenado Manual
Ingresa Matricula: 372688
Ingresa Apellido Paterno: quiros
Ingresa Apellido Materno: vicencio
Ingresa Nombre: diego demian
Ingresa Edad: 19
Ingresa sexo 1] Hombre 2] Mujer: 1
Presione una tecla para continuar . . .
```

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

Ingresa funcion a utilizar:

6

Matricula	ApPat	ApMat	Nombre	Edad	Sexo
331119	Perez	Aguilar	Juan Guillermo	32	Hombre
304243	Ruiz	Rivera	Guillermo Carlos	22	Hombre
314810	Gonzalez	Perez	Manuel	18	Hombre
307040	Ramirez	Gonzalez	Vanessa Sofia	29	Mujer
321203	Jimenez	Perez	Alicia Beatriz	25	Mujer
320905	Diaz	Vicencio	Laura Julieta	40	Mujer
317174	Perez	Lopez	Natalia Carolina	28	Mujer
300900	Martinez	Vargas	Luis Miguel	39	Hombre
310543	Vargas	Lopez	Lorena	33	Mujer
323721	Parra	Castro	Alejandro Jorge	30	Hombre
300343	Torres	Sanchez	Elena	37	Mujer
326532	Vargas	Morales	Daniel Manuel	36	Hombre
314833	Ruiz	Sanchez	Marta	17	Mujer
330443	Ramirez	Navarro	Vanessa	37	Mujer
300672	Lopez	Perez	Carolina	27	Mujer
316783	Ortega	Rodriguez	Alejandro	20	Hombre
311680	Vicencio	Jimenez	Juan	28	Hombre
309474	Vicencio	Rivera	Maria	40	Mujer
331110	Ruiz	Ramirez	Sergio	33	Hombre
302757	Aguilar	Castro	Raul	21	Hombre
372688	Quiros	Vicencio	Diego Demian	19	Hombre

Presione una tecla para continuar . . .

R E G I S T R O S U A B C

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

Ingresa funcion a utilizar:

4

Ingresa matricula a buscar: 300900

Matricula	ApPat	ApMat	Nombre	Edad	Sexo
300900	Martinez	Vargas	Luis Miguel	39	Hombre

Presione una tecla para continuar . . .

- R E G I S T R O S U A B C
- 1.- AGREGAR (AUTOM 10 REGISTROS)
 - 2.- AGREGAR MANUAL
 - 3.- ELIMINAR REGISTRO
 - 4.- BUSCAR
 - 5.- ORDENAR
 - 6.- IMPRIMIR
 0. Salir

Ingresa funcion a utilizar:

5

Registros Actualmente ordenados

Presione una tecla para continuar . . .

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

Ingresa funcion a utilizar:

6

Matricula	ApPat	ApMat	Nombre	Edad	Sexo
300343	Perez	Aguilar	Juan Guillermo	32	Hombre
300672	Ruiz	Rivera	Guillermo Carlos	22	Hombre
300900	Gonzalez	Perez	Manuel	18	Hombre
302757	Ramirez	Gonzalez	Vanessa Sofia	29	Mujer
304243	Jimenez	Perez	Alicia Beatriz	25	Mujer
307040	Diaz	Vicencio	Laura Julieta	40	Mujer
309474	Perez	Lopez	Natalia Carolina	28	Mujer
310543	Martinez	Vargas	Luis Miguel	39	Hombre
311680	Vargas	Lopez	Lorena	33	Mujer
314810	Parra	Castro	Alejandro Jorge	30	Hombre
314833	Torres	Sanchez	Elena	37	Mujer
316783	Vargas	Morales	Daniel Manuel	36	Hombre
317174	Ruiz	Sanchez	Marta	17	Mujer
320905	Ramirez	Navarro	Vanessa	37	Mujer
321203	Lopez	Perez	Carolina	27	Mujer
323721	Ortega	Rodriguez	Alejandro	20	Hombre
326532	Vicencio	Jimenez	Juan	28	Hombre
330443	Vicencio	Rivera	Maria	40	Mujer
331110	Ruiz	Ramirez	Sergio	33	Hombre
331119	Aguilar	Castro	Raul	21	Hombre
372688	Quiros	Vicencio	Diego Demian	19	Hombre

Presione una tecla para continuar . . .

```
R E G I S T R O S U A B C
1.- AGREGAR (AUTOM 10 REGISTROS)
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0. Salir
```

Ingresar función a utilizar:

4

Ingresar matrícula a buscar: 372688

Matricula	ApPat	ApMat	Nombre	Edad	Sexo
372688	Quiros	Vicencio	Diego Demian	19	Hombre

Presione una tecla para continuar . . .

Ingresar matrícula a dar de baja: 310543

Alumno dado de baja.

Presione una tecla para continuar . . .

```
R E G I S T R O S U A B C
1.- AGREGAR (AUTOM 10 REGISTROS)
2.- AGREGAR MANUAL
3.- ELIMINAR REGISTRO
4.- BUSCAR
5.- ORDENAR
6.- IMPRIMIR
0. Salir
```

Ingresar función a utilizar:

0

Fin del programa