



Universidad Autónoma de Baja California
Facultad de Ingeniería Arquitectura y Diseño



Ingeniero en Software y Tecnologías Emergentes.

Programación estructurada

ARCHIVOS

Actividad 12

Danna Guadalupe Sandez Islas

373080

1.- Cargar Archivo

```
case 1: // cargar registros desde el arch de tecto
    if (!carg)
    {
        gen_arch(registro, i);
    }
    else
    {
        printf("Los registros ya se han cargado\n");
    }
    system("PAUSE");
    break;
```

```
void gen_arch(Tdatos registro[], int i)
{
    int carg = 0;
    char nom_arch[20];
    nom(nom_arch);
    carg = carg_arch(nom_arch, registro, &i);

    if (carg)
    {
        printf("Registros cargados al .txt con exito!\n");
    }
    else
    {
        printf("Parece que ha ocurrido un problema \n");
    }
}
```

Salida:

```
Ingresa el nombre del archivo: datox
Registros cargados al .txt con exito!
Presione una tecla para continuar . . .
```

2.- Agregar

```
case 2: // agregar registros al arreglo y al arch de txt
if (i < 1500)
{
    for (int j = 0; j < 10; j++)
    {
        persona = gen_per();
        while (busq_seq(registro, i, persona.mat) != -1)
        {
            persona.mat = numAleatorio(300000, 399999);
        }
        registro[l++] = persona;
        i++;
    }
    ord = 0;
    printf("10 personas agregadas exitosamente!\n");
}
else
{
    printf("El registro a llegado a su capacidad maxima!\n");
}

if (carg)
{
    escr_arch(nom_arch, registro, i, 0);
    escr_arch(nom_arch, registro, i, 1);
}
system("PAUSE");
break;
```

```

Tdatos gen_per(void)
{
    Tdatos per;
    per.status = 1;
    per.mat = numAleatorio(300000, 399999);
    strcpy(per.ap_pat, apellido[rand() % 30]);
    strcpy(per.ap_mat, apellido[rand() % 30]);

    int sex;
    sex = numAleatorio(1, 2);
    if (sex == 1)
    {
        strcpy(per.nom, nom_mujer[rand() % 19]);
        strcpy(per.sexo, "MUJER");
    }
    else
    {
        strcpy(per.nom, nom_hombre[rand() % 19]);
        strcpy(per.sexo, "HOMBRE");
    }

    per.edad = numAleatorio(18, 30);
    return per;
}

```

```

void escr_arch(char nomArchivo[], Tdatos vect[], int n, int arch)
{
    int i, cont = 0;
    FILE *fa;

    char temp[30];
    strcpy(temp, nomArchivo);

    if (!arch)
    {
        strcat(temp, "_activos");
    }

    strcat(temp, ".txt");
    fa = fopen(temp, "w");

    if (arch)
    {
        fprintf(fa, "-----\n");
        fprintf(fa, "  No  | MATRICULA | NOMBRE      | APELLIDO P.  | APELLIDO MAT.  | EDAD | SEXO \n");
        fprintf(fa, "-----\n");
    }

    for (i = 0; i < n; i++)
    {
        if (vect[i].status == 1)
        {
            fprintf(fa, "%4d.- %6d    %-10s    %-10s    %-10s    %2d    %-7s", cont, vect[i].mat, vect[i].nom, vect[i].ap_pat, vect[i].ap_m,
            if (i < n - 1)
            {
                fprintf(fa, "\n");
            }
            cont++;
        }
    }

    fclose(fa);
}

```

Salida:

```
10 personas agregadas exitosamente!  
Presione una tecla para continuar . . . █
```

3.- Eliminar

```
case 3: // eliminar  
    eliminar(registro, i, ord);  
    system("PAUSE");  
    break;
```

```
void eliminar(Tdatos registro[], int i, int ord)  
{  
    int num, k = 0, op;  
    num = validar("Ingrese el numero de matricula que desea eliminar: ", 300000, 399999);  
    system("CLS");  
    k = busq_opt(registro, i, num, ord);  
    if (k != -1)  
    {  
        if (registro[k].status == 1)  
        {  
            printf("Desea eliminar a %d?\n1-Si\n2-No\n", registro[k].mat);  
            op = validar("Ingrese una opcion: ", 1, 2);  
            system("CLS");  
            if (op == 1)  
            {  
                registro[k].status = 0;  
                printf("La matricula eliminada exitosamente!\n");  
            }  
            else  
            {  
                printf("La matricula no se ha eliminado\n");  
            }  
        }  
        else  
        {  
            printf("Matricula inactiva\n");  
        }  
    }  
    else  
    {  
        printf("La matricula no es existente\n");  
    }  
}
```

salida:

```
Matricula inactiva  
Presione una tecla para continuar . . . █
```

4.- Buscar

```
case 4: // buscar
    num = validar("Ingrese la matricula a buscar: ", 300000, 399999);
    system("CLS");
    k = busq_opt(registro, i, num, ord);
    if (k != -1)
    {
        printf("MATRICULA: %d\nNOMBRE: %s\nAPELLIDO PATERNO: %s\nAPELLIDO MATERNO: %s\nEDAD: %d\nSEXO: %s\n", registro[k].mat, registro[k].nom, registro[k].ap_pat
    }
    else
    {
        printf("Matricula no encontrada\n");
    }
    system("PAUSE");
    break;
```

```
int busq_opt(Tdatos refistro[], int n, int mat, int flag)
{
    int i;

    if (flag)
    {
        i = busq_binaria(refistro, n, mat);
    }
    else
    {
        i = busq_seq(refistro, n, mat);
    }

    return i;
}
```

```
int busq_binaria(Tdatos registro[], int rf, int mat)
{
    int ri = 0;
    while (ri <= rf)
    {
        int medium = ri + (rf - ri) / 2;

        // Checa si el numero se encuentra en medio
        if (registro[medium].mat == mat)
        {
            return medium;
        }

        if (registro[medium].mat < mat) // si el numero es mayor ignora el lado izquierdo
        {
            ri = medium + 1;
        }
        else // si el numero es menor, ignora el lado derecho
        {
            rf = medium - 1;
        }
    }
    // el numero no se encuentra
    return -1;
}
```

```
int busq_seq(Tdatos registro[], int n, int num) // registro no ordenado
{
    int i;
    for (i = 0; i < n; i++)
    {
        if (registro[i].mat == num)
        {
            return i;
        }
    }
    // no encontrado
    return -1;
}
```

MATRICULA: 311858

NOMBRE: CARLOS

APELLIDO PATERNO: RAMIREZ

APELLIDO MATERNO: MENDOZA

EDAD: 26

SEXO: HOMBRE

Presione una tecla para continuar . . . █

5.- Ordenar

```
case 5: // ordenar
    if (ord == 1)
    {
        printf("El registro ya se encuentra ordenado\n");
    }
    else
    {
        ord = ordOpt(registro, i);
        printf("Registro ordenado con exito1\n");
    }
    system("PAUSE");
    break;
```

```
int ordOpt(Tdatos registro[], int n)
{
    if (n <= 200)
    {
        burbuja(registro, n);
    }
    else
    {
        quickSort(registro, n);
    }

    return 1;
}
```



```
int burbuja(Tdatos registro[], int n)
{
    int i, j;
    Tdatos temp;
    for (i = 0; i < n; i++)
    {
        for (j = i + 1; j < n; j++)
        {
            if (registro[j].mat < registro[i].mat)
            {
                temp = registro[i];
                registro[i] = registro[j];
                registro[j] = temp;
            }
        }
    }
    return 1;
}
```

```
int quickSort(Tdatos vect[], int n)
{
    qs(vect, 0, n - 1);

    return 1;
}
```

int limi

```

void qs(Tdatos vect[], int limite_izq, int limite_der)
{
    int izq, der, pivote;
    Tdatos temp;

    izq = limite_izq;
    der = limite_der;
    pivote = vect[(izq + der) / 2].mat;

    do
    {
        while (vect[izq].mat < pivote && izq < limite_der)
        {
            izq++;
        }

        while (pivote < vect[der].mat && der > limite_izq)
        {
            der--;
        }

        if (izq <= der)
        {
            temp = vect[izq];
            vect[izq] = vect[der];
            vect[der] = temp;
            izq++;
            der--;
        }
    } while (izq <= der);

    if (limite_izq < der)
    {
        qs(vect, limite_izq, der);
    }

    if (limite_der > izq)
    {
        qs(vect, izq, limite_der);
    }
}

```

Salida:

```

Registro ordenado con exito1
Presione una tecla para continuar . . . █

```

7.- Generar Archivo

```
case 7: // generar archivo
    nom(nom_arch);
    escr_arch(nom_arch, registro, i, 0);
    escr_arch(nom_arch, registro, i, 1);
    printf("Tarea realizada exitosamente!\n");
    system("PAUSE");
    break;
```

```
void nom(char nom_arch[])
{
    do
    {
        system("CLS");
        printf("Ingresa el nombre del archivo: ");
        fflush(stdin);
        gets(nom_arch);
    } while (strcmp(nom_arch, "\n") == 0);
}
```

```
void escr_arch(char nomArchivo[], tdatos vect[], int n, int arch)
{
    int i, cont = 0;
    FILE *fa;

    char temp[30];
    strcpy(temp, nomArchivo);

    if (larch)
    {
        strcat(temp, "_activos");
    }

    strcat(temp, ".txt");
    fa = fopen(temp, "w");

    if (arch)
    {
        fprintf(fa, "-----\n");
        fprintf(fa, " No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
        fprintf(fa, "-----\n");
    }

    for (i = 0; i < n; i++)
    {
        if (vect[i].status == 1)
        {
            fprintf(fa, "%4d.- %6d %-10s %-10s %-10s %2d %-7s", cont, vect[i].mat, vect[i].nom, vect[i].ap_pat, vect[i].ap_m, vect[i].edad, vect[i].sexo);
            if (i < n - 1)
            {
                fprintf(fa, "\n");
            }
            cont++;
        }
    }

    fclose(fa);
}
```

Salida:

```
Ingresa el nombre del archivo: datox
Tarea realizada exitosamente!
Presione una tecla para continuar . . . █
```

8.- Cantidad de registros en Archivo

```
break;
case 8: // retorno de cantidad de datos existentes en el archivo
    st = validar("Que registros desea ver?\n1-Activos\n2-Eliminados\nIngrese una opcion: ", 1, 2);
    nom(nom_arch);
    num = cont_reg(nom_arch, st);
    system("CLS");
    if (num == -1)
    {
        printf("El archivo no se ha encontrado\n");
    }
    else
    {
        printf("%d registros existentes en el archivo\n", num);
    }
    system("PAUSE");
    break;
```

```
int cont_reg(char nomArchivo[], int status)
{
    int cont;
    char cmd[50];

    system("mingw32-gcc-6.3.0.exe cont_reg.c -o cont_reg");
    system("PAUSE");
    sprintf(cmd, "cont_reg.exe %s %d", nomArchivo, status);
    cont = system(cmd);

    return cont;
}
```

Salida:

```
19 registros existentes en el archivo
Presione una tecla para continuar . . . █
```

9.- Mostrar Borrados

```
case 9: // mostrar registros eliminados
    imprimir(registro, i, 0);
    system("PAUSE");
    break;
```

```
void imprimir(tdatos vect[], int n, int status)
{
    int activos, i;

    printf("-----\n");
    printf("No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO \n");
    printf("-----\n");
    for (i = 0, activos = 0; i < n; i++)
    {
        if (vect[i].status == status)
        {
            printf("%4d.- %6d %10s %10s %10s %2d %-7s\n", activos + 1, vect[i].mat, vect[i].nom, vect[i].ap_pat, vect[i].ap_mat, vect[i].
            activos++;
        }
    }
}
```

```
-----
No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO
-----
1.- 303208 ANA GONZALEZ GUTIERREZ 24 MUJER
Presione una tecla para continuar . . .
```

Archivos .txt:

Danna_Sandez_Programacion_Estructurada > Actividad_12 > output > datox.txt

```
1 -----
2 No | MATRICULA | NOMBRE | APELLIDO P. | APELLIDO MAT. | EDAD | SEXO
3 -----
4 0.- 300763 LUIS GARCIA MEZA 20 HOMBRE
5 1.- 301605 ANGEL CASTILLO LOPEZ 18 HOMBRE
6 2.- 303610 ALEJANDRO MENDOZA FLORES 18 HOMBRE
7 3.- 303766 FERNANDO LOPEZ SAAVEDRA 23 HOMBRE
8 4.- 304056 MANUEL RUIZ SANDEZ 28 HOMBRE
9 5.- 307125 ANDRES RODRIGUEZ TORRES 19 HOMBRE
10 6.- 309352 BEATRIZ HERNANDEZ TORRES 26 MUJER
11 7.- 309789 ANTONIO RAMIREZ MENDOZA 24 HOMBRE
12 8.- 311858 CARLOS RAMIREZ MENDOZA 26 HOMBRE
13 9.- 313338 MANUEL SANCHEZ HERNANDEZ 28 HOMBRE
14 10.- 315836 MIGUEL GARCIA MENDOZA 23 HOMBRE
15 11.- 316243 PATRICIA HARO SOLANO 22 MUJER
16 12.- 323144 ANTONIO TORRES SAAVEDRA 27 HOMBRE
17 13.- 324344 LAURA ROJAS MENDOZA 30 MUJER
18 14.- 326014 ANDRES MARTINEZ LOPEZ 26 HOMBRE
19 15.- 326498 FERNANDO GARCIA PEREZ 28 HOMBRE
20 16.- 328843 LUIS TORRES MEZA 22 HOMBRE
21 17.- 331400 JOSE RAMIREZ VARGAS 21 HOMBRE
22 18.- 332599 ANTONIO SANDEZ RODRIGUEZ 23 HOMBRE
```

1	0.-	300763	LUIS	GARCIA	MEZA	20	HOMBRE
2	1.-	301605	ANGEL	CASTILLO	LOPEZ	18	HOMBRE
3	2.-	303610	ALEJANDRO	MENDOZA	FLORES	18	HOMBRE
4	3.-	303766	FERNANDO	LOPEZ	SAAVEDRA	23	HOMBRE
5	4.-	304056	MANUEL	RUIZ	SANDEZ	28	HOMBRE
6	5.-	307125	ANDRES	RODRIGUEZ	TORRES	19	HOMBRE
7	6.-	309352	BEATRIZ	HERNANDEZ	TORRES	26	MUJER
8	7.-	309789	ANTONIO	RAMIREZ	MENDOZA	24	HOMBRE
9	8.-	311858	CARLOS	RAMIREZ	MENDOZA	26	HOMBRE
10	9.-	313338	MANUEL	SANCHEZ	HERNANDEZ	28	HOMBRE
11	10.-	315836	MIGUEL	GARCIA	MENDOZA	23	HOMBRE
12	11.-	316243	PATRICIA	HARO	SOLANO	22	MUJER
13	12.-	323144	ANTONIO	TORRES	SAAVEDRA	27	HOMBRE
14	13.-	324344	LAURA	ROJAS	MENDOZA	30	MUJER
15	14.-	326014	ANDRES	MARTINEZ	LOPEZ	26	HOMBRE
16	15.-	326498	FERNANDO	GARCIA	PEREZ	28	HOMBRE
17	16.-	328843	LUIS	TORRES	MEZA	22	HOMBRE
18	17.-	331400	JOSE	RAMIREZ	VARGAS	21	HOMBRE
19	18.-	332599	ANTONIO	SANDEZ	RODRIGUEZ	23	HOMBRE