



## Problemas Resueltos

### EJEMPLO 01 Manejo de Archivos Binarios con datos de NÚMEROS ENTEROS.

#### OBJETIVO:

- Apertura y cierre de archivo.
- Pasar por parámetro el archivo.
- Ingreso de datos.
- Recorrido y listado de un archivo.
- Operaciones de cálculo usando los datos del Archivo.
- Eliminación de un valor del archivo (Para estos casos se usa un vector auxiliar).

#### ENUNCIADO:

Crear un archivo binario (Multiplo.Dat) e ingresar una serie indeterminada de números entero.

#### SE PIDE:

- Por medio de una función realizar la carga de los datos teniendo en cuenta que se:
  - Deberá guardar en el archivo solo aquellos valores que sean múltiplos de 3.
  - Terminada la carga se deberá mostrar en la función main, la cantidad de valores que no se guardaron.
- Usando una función, Deberá mostrar por pantalla, los valores que están guardado en el archivo, cada valor separado por dos guiones medio y como máximo 10 valores por fila.
- En una función se deberá buscar un determinado valor en el archivo. El número a buscar debe ser ingresado en la función main y pasado como parámetro para determinar la cantidad de veces que se guardó el valor.

#### CODIFICACIÓN:

```
#include <stdio.h>
#include <stdlib.h>

int ingresoDeValoresEnteros(FILE *archMultiplo, int &cantidadIngresada);
void listadoDeArchivo(FILE *archMultiplo);
int buscarUnValor(FILE *archMultiplo, int numeroQueBusca);

main()
{
    FILE *archMultiplo;
    archMultiplo = fopen("Multiplo.dat", "w+b");
    if(archMultiplo == NULL)
    {
        system("CLS");
        printf("\n\n Se produjo un ERROR al intentar abrir el archivo\n");
        printf("comuniquese con el administrador del Sistema. Gracias");
        printf("\n\n\t");
        system("PAUSE");
        exit(1);
    }
    int cantidadNumerosNoGuardados=0, cantidadIngresada=0;
    cantidadNumerosNoGuardados = ingresoDeValoresEnteros(archMultiplo, cantidadIngresada);
    system("COLOR 4F"); //Establece color de fondo rojo y letras blancas.
```

Define una variable de tipo puntero a archivo.

Abre el archivo de Escritura-.

Verifica si el archivo se abrió mal o no se abrió muestra un mensaje y cierra el programa.

```

printf("\n\t\t*****");
printf("\n\t\t** DE LOS %2d VALORES INGRESADOS NO SE **", cantidadIngresada);
printf("\n\t\t** GUARDARON %2d POR NO SER MULTIPLOS DE 3 **", cantidadNumerosNoGuardados);
printf("\n\t\t*****\n");
system("PAUSE");
system("CLS");

listadoDeArchivo(archMultiplo);

system("CLS");
printf("\n\n\n\t\t");
printf("Ingrese el Valor que desea Buscar: ");
int valorQueBusca=0;
scanf("%d", &valorQueBusca);
int cantidadDeVeces=0;
cantidadDeVeces = buscarUnValor(archMultiplo, valorQueBusca);
printf("\n\n\n");
printf("\n\t\t\t*****");
printf("\n\t\t\t** El valor buscado (%3d) Se ingreso al archivo %3d veces **", valorQueBusca,
    cantidadDeVeces);
printf("\n\t\t\t*****");
printf("\n\n\n");
system("PAUSE");
fclose(archMultiplo); //cierra el archivo.
}
/*-----
Nombre      : ingresoDeValoresEnteros(...)
Descripción : De los valores ingresados Guarda solo aquellos que sean múltiplos de 3 y cuenta cuantos valores NO se guardaron.
Tipo Función : con Tipo.
Retorna     : variable de tipo entero indicando cuantos datos no se guardaron.
Parámetros  : Recibe dos
                1- El puntero del registro.
                2- Una variable por referencia que indica cuantos datos se ingresaron.
-----*/
int ingresoDeValoresEnteros(FILE *archMultiplo, int &cantidadIngresada)
{
    int valorEntero, cantidadNoGuardada=0;
    char opcSalida = 'N';
    system("COLOR 30"); //color de fondo Aguamarina y letra negra.
    do{
        printf("\nIngrese un Nro Entero: ");
        scanf("%d", &valorEntero);

        cantidadIngresada++;
        if ( valorEntero%3 == 0) //Determina si es múltiplo de 3
        {
            fwrite(&valorEntero, sizeof(valorEntero), 1, archMultiplo); //Guarda el valor ingresado
        }else{
            cantidadNoGuardada++; //Cuenta los no guardados
        }
        printf("\n\t Continua: (S)i - (N)o, Ingrese Opcion: "); //Pregunta si continua
        _flushall();
        scanf("%c", &opcSalida); //recibe la respuesta por teclado.
    }while ((opcSalida == 'S') || (opcSalida == 's')); //Repite mientras "opcSalida" sea S o s?

    printf("\n\n\n");
    printf("\n\t\t\t*****");
    printf("\n\t\t\t** F I N A L D E L A C A R G A **");
    printf("\n\t\t\t*****");
    printf("\n\n\n");
    system("PAUSE");
    system("CLS");
    return cantidadNoGuardada;
}

```

```

/*-----
Nombre      : listadoDeArchivo(...)
Descripción : Lista los datos del archivo que recibe por parámetro
Tipo Función : Sin Tipo.
Retorna     : Ningun Valor.
Parámetros  : Recibe
                1- El puntero al registro.
-----*/
void listadoDeArchivo(FILE *archMultiplo)
{
    int numeroGuardado, cantidadDeColumna;
    cantidadDeColumna = 1;

    printf("\n\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t**      L I S T A D O   D E L   A R C H I V O      **");
    printf("\n\t\t*****");
    printf("\n\n\n");

    rewind(archMultiplo); //Ubica el puntero en el primer valor que tiene el archivo.

    fread(&numeroGuardado, sizeof(int), 1, archMultiplo); //Lee un valor.
    printf("\t\t");
    while ( !feof(archMultiplo) ) //Mientras no se fin de archivo hacer
    {
        printf("%03d -- ", numeroGuardado);
        if(cantidadDeColumna==3)
        {
            printf("\n\t\t");
            cantidadDeColumna=1;
        }else
            cantidadDeColumna++;
        fread(&numeroGuardado, sizeof(int), 1 , archMultiplo);
    }
    printf("\n\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t**      F I N A L       D E L   L I S T A D O      **");
    printf("\n\t\t*****");
    printf("\n\n");
    system("PAUSE");
}

/*-----
Nombre      : buscarUnValor(...)
Descripción : Borra del archivo el numero que recibe por parametro.
Tipo Función : Con Tipo.
Retorna     : Cantidad de veces que se encontro el valor.
Parámetros  : Recibe
                1- El puntero del registro.
                2- El numero que debe buscar.
-----*/
int buscarUnValor(FILE *archMultiplo, int numeroQueBusca)
{
    int cantidadDeVeces = 0, numeroGuardado;

    rewind(archMultiplo); //Ubica el puntero en el primer valor que tiene el archivo.

    fread(&numeroGuardado, sizeof(int), 1, archMultiplo); //Lee el primer valor.
    while ( !feof(archMultiplo) ) //Mientras no se fin de archivo hacer
    {
        if(numeroGuardado == numeroQueBusca)
            cantidadDeVeces++;
        fread(&numeroGuardado, sizeof(int), 1 , archMultiplo);
    }
    return cantidadDeVeces;
}

```

**OBJETIVO:**

- + Apertura y cierre de archivo.
- + Pasar por parámetro el archivo.
- + Ingreso de datos.
- + Recorrido y listado de un archivo.
- + Operaciones de cálculo usando los datos del Archivo.

**ENUNCIADO:**

La empresa Distribuidora *Cielo Abierto* almacenas el total de todas las ventas realizadas durante el día. Al finalizar el día

**SE PIDE:**

- 4- Haciendo uso de una función sin tipo, realizar la carga de las diferentes ventas y guardar en un archivo binario de números reales. La función deberá recibir por parámetro el Archivo.
- 5- Haciendo uso de una función con tipo, recorrer el archivo determinando la cantidad valores superiores o iguales a los \$ 1000 y por otro lado, las ventas menores a esta cifra. **Mostrar ambos resultado en la función main.** La función deberá recibir por parámetro el archivo binario.
- 6- Realizar el **listado** del archivo, donde las diferentes cantidades deben estar separada por un guion medio y mostrando 10 cantidades por fila.

**CODIFICACIÓN:**

```
#include <stdio.h>
#include <stdlib.h>

/*----- Prototipos de Funciones -----*/
void cargarArchivo(FILE *arch1);
int contarMayorMenorQue(FILE *arch1, int &menorQue);
void listarArchivo(FILE *arch1);
/*----- Bloque Principal -----*/
main (void)
{
    FILE *arch; //Define "arch" de tipo puntero a un archivo.
    int menorQue=0, mayorQue=0; //Define los contador de Valores mayores y menores que 1000.

    /*----- Abre y Determina si se creó bien el archivo ----*/
    arch = fopen("ventas.dat", "w+b");
    if(arch==NULL){
        printf("\n\n Ocurrió un error al intentar abrir el archivo....\n");
        system("PAUSE");
        exit(1);
    }

    /*----- Realiza la carga del archivo -----*/
    cargarArchivo(arch);

    /*----- Cuenta la cantidad de ventas mayores y menores a 1000 -----*/
    mayorQue = contarMayorMenorQue(arch, menorQue);

    /*----- Muestra los valores obtenidos -----*/
    printf("Hay %d Ventas con importe inferiores a $1000 \n\n", menorQue);
    printf("Hay %d Ventas con importe superiores o iguales a $1000\n\n", mayorQue);
    system("PAUSE");
}
```

Crea y abre un nuevo archivo de Escritura, Si existe un archivo con ese nombre lo crea de nuevo

En caso que el archivo no se halla creado por algún motivo, mostrara un mensaje y cierra el programa.

```

system("CLS");
/*----- Realiza el listado del archivo -----*/
listarArchivo(arch);
/*-----Cierra el archivo -----*/
fclose(arch); //Cierra el archivo.
} //Fin función main.

/*-----
Nombre      : cargarArchivo(...)
Descripción : Realiza la carga de las Ventas.
Tipo Función : Sin Tipo.
Retorna     : Ningún Valor.
Parámetros  : Recibe uno
              1- El puntero del registro.
-----*/
void cargarArchivo(FILE *arch1)
{
    float montoVenta;
    char seguir = 'n'; //Define "Seguir" para determinar el final de la carga.
    printf("\n\n\n*****\n");
    printf("\n          Inicio de la Carga          ");
    printf("\n\n*****\n\t");
    do{
        system("CLS");
        printf("Importe:  ");
        scanf("%f", &montoVenta);
        fwrite(&montoVenta, sizeof(float), 1, arch1); //Graba el monto en el archivo.
        _flushall();
        printf("\nIngresar mas datos (s/n): ");
        scanf("%c", &seguir);
    } while( seguir == 'S' || seguir == 's' );
    printf("\n\n\n*****\n");
    printf("\n          Fin de la Carga          ");
    printf("\n\n*****\n\t");
    system("PAUSE");
    system("CLS");
}

/*-----
Nombre      : contarMayorMenorQue(...)
Descripción : Cuenta las ventas mayores o iguales a 1000 y las menores a 1000.
Tipo Función : Sin Tipo.
Retorna     : un valor de tipo entero.
Parámetros  : Recibe dos
              1- El puntero al archivo.
              2- Variables por referencia. Contador de las Ventas menores a 1000.
-----*/
int contarMayorMenorQue(FILE *arch1, int &menorQue)
{
    int mayorQue=0;
    float venta;
    rewind(arch1); //Se ubica al inicio del archivo.
    fread(&venta, sizeof(venta), 1, arch1); //Leer el primer valor.
    menorQue = 0;
    mayorQue = 0;
    while( !feof(arch1)) //Termina después de leer el último valor.
    {
        if (venta < 1000)
            menorQue = menorQue + 1;
        else
            mayorQue++;
    }
}

```



**OBJETIVO:**

- + Apertura y cierre de archivo.
- + Pasar por parámetro el archivo y Registro.
- + Ingreso de datos.
- + Recorrido y listado de un archivo.
- + Cerrar y Abrir el archivo en diferentes Formas. (Nuevo, Agregar, Lectura)

**ENUNCIADO:**

En un negocio se registran los totales de ventas de los **N** artículos que se vendieron durante el día, en un archivo Binario “**ventasDiarias.dat**”. Los datos que se guardan son los siguientes:

**Configuración lógica del registro:**

- código Artículo      N(6).
- Descripción          C(50).
- importe de total de la venta diaria Venta   R(3,5).

**SE PIDE:**

- a) Definir de manera apropiada el registro y **realizar la carga de las diferentes ventas** por medio de una función Sin Tipo. La función recibe por parámetro el puntero al archivo y la cantidad de Artículos que registraron ventas durante el día.
- b) Usando una función Con Tipo, recorrer el archivo determinando la cantidad de ventas superiores a los \$ 1000. Mostrar el resultado en main.
- c) Usando una función Sin Tipo, Realizar el listado del archivo.

**CODIFICACIÓN:**

```
#include <stdio.h>
#include <stdlib.h>

/*--- Estructura Lógica del Registro Venta ---*/
struct ventaDiaria{
    int codigoArticulo;
    char nombreDelArticulo[50];
    float totalVentaDiaria;
};

/*--- Prototipos de Funciones ---*/
void registrarVentaDiaria(FILE *arch1, int cantArt);
int contarMayorQue(FILE *arch1);
void listarArchivo(FILE *arch1);

/*--- Bloque Principal ---*/
main (void)
{
    FILE *arch; // Crea una variable "arch" de tipo puntero a un archivo.
    int mayorQue1000=0, cantidadArticulos=0;
    printf("Ingresar la cantidad de Articulos que registraron ventas el dia de hoy: ");
    scanf("%d", &cantidadArticulos);

    /*---- Realiza la carga del archivo ----*/
    registrarVentaDiaria(arch, cantidadArticulos);

    /*---- Cuenta la cantidad de ventas mayores a 1000 ----*/
    mayorQue1000 = contarMayorQue(arch);

    /*---- Muestra los valores obtenidos ----*/
```

```

printf("Hay %3d productos con importe mayores a $1000\n", mayorQue1000);
system("PAUSE");
system("CLS");

/*--- Realiza el listado del archivo ---*/
listarArchivo(arch); //Llamada
} //Fin de función main.

/*-----
Nombre      : registrarVentaDiaria(...)
Descripción : Realiza la carga de los datos en el archivo binario.
Tipo Función : Sin Tipo.
Retorna     : --
Parámetros  : recibe dos
               1- Variable tipo puntero a archivo.
               2- Cantidad de artículos a guardar en el archivo.
-----*/

void registrarVentaDiaria(FILE *arch1, int cantArt)
{
    ventaDiaria regVenta; //Variable "regVenta" de tipo Venta(estructura lógica).

    /*--- Abre Archivo ---*/
    arch1 = fopen("ventasDiarias.dat", "wb"); //Crea el archivo de Escritura.

    /*--- Realiza la carga de datos ---*/
    printf("\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t** INICIO DE CARGA DE VENTAS **");
    printf("\n\t\t*****");
    printf("\n\n");
    for(int i=1; i<=cantArt; i++)
    {
        printf("\n");
        printf("\n\t*****");
        printf("\n\t\t** Articulo Nro: %03d **", i); //recibe 3 valores y rellena con cero a la izquierda.
        printf("\n\t*****");
        printf("\n\n");
        printf("\n\t%17s", "Codigo : "); scanf(" %3d", &regVenta.codigoArticulo);
        _flushall();
        printf("\n\t%17s", "Nombre Articulo: "); gets(regVenta.nombreDelArticulo);
        printf("\n\t%17s", "Importe: "); scanf("%6.2f", &regVenta.totalVentaDiaria);
        fwrite(&regVenta, sizeof(ventaDiaria), 1, arch1); //Graba el registro.
        _flushall();
    }
    printf("\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t** FIN DE LA CARGA VENTAS **");
    printf("\n\t\t*****");
    printf("\n\n\t");
    system("PAUSE");
    system("CLS");

    fclose(arch1); //Cierra el archivo
}

/*-----
Nombre : contarMayorQue(...)
Descripción : Cuenta las ventas mayores e iguales a 1000 y las menores a 1000.
Tipo Función : Con Tipo.
Retorna : El cantidad de valores que fueron menores a 1000 y por referencia mayor de 1000
Parámetros : Recibe dos (1)- El puntero del registro. (2)- Variable por referencia. Ventas mayores a 1000.
-----*/

```



```

int contarMayorQue(FILE *arch1)
{
    ventaDiaria regVenta;
    int mayorQue = 0;

    arch1 = fopen("ventasDiarias.dat", "rb"); // Abre el archivo de solo lectura

    fread(&regVenta, sizeof(ventaDiaria), 1, arch1); //Leer el primer registro.

    /*--- Lista el archivo hasta el final ---*/
    while( !feof(arch1)) //Termina después de leer el último registro.
    {
        if (regVenta.totalVentaDiaria > 1000.00)
            mayorQue++;

        fread(&regVenta, sizeof(ventaDiaria), 1, arch1); //Continua leyendo.
    }

    fclose(arch1); //Cierra el archivo

    return mayorQue;
}

/* -----
Nombre : listarArchivo(...)
Descripción : Lista todas las ventas Realizadas (registradas).
Tipo Función : Sin Tipo.
Retorna : Ningún Valor.
Parámetros : Recibe el puntero del registro.
----- */
void listarArchivo(FILE *arch1)
{
    ventaDiaria regVenta; //crea la variable "regVenta" de tipo venta.
    arch1 = fopen("ventasDiarias.dat", "rb"); // Abre el archivo de solo lectura
    fread(&regVenta, sizeof(ventaDiaria), 1, arch1); //Lee un registro.

    printf("\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t** INICIO DEL LISTADO **");
    printf("\n\t\t*****");
    printf("\n\t\t** Codigo Nombre del Articulo Monto **");
    printf("\n\t\t*****");
    while(!feof(arch1)) //Termina después de leer el último registro.
    {
        printf("\n\t\t%-10d%-40s%-6.2f", regVenta.codigoArticulo, regVenta.nombreDelArticulo, regVenta.totalVentaDiaria);
        fread(&regVenta, sizeof(ventaDiaria), 1, arch1); //Continua leyendo.
    }
    printf("\n\n");
    printf("\n\t\t*****");
    printf("\n\t\t** FIN DEL LISTADO **");
    printf("\n\t\t*****");
    printf("\n\n\t\t");

    system("PAUSE");
    system("CLS");
    fclose(arch1); //Cierra el archivo
}

```

**OBJETIVO:** *Manejo de archivo Binario de registro.*

- + Guardar datos de un registro a un archivo binario (Alta)
- + Eliminación de los datos de un registro (Baja Física).
- + Trabajo con registro jerarquizado.
- + Búsqueda de dato en el archivo de registro
- + Borrado de un archivo.
- + Pasar información de un archivo a otro.
- + Cambiar nombre de un archivo.

**ENUNCIADO:**

En una empresa de taxis se registra la información en un archivo binario llamado Taxi.dat, que posee la siguiente configuración de registro:

- + Número de licencia.
- + Apellido y Nombre del Chofer
- + Fecha de Incorporación o compra del vehículo por la empresa. (*Registro Jerarquizado*).

**SE PIDE:**

- a) *Crear la estructura* adecuada. Y utilizando un menú de opciones realizar
- b) *Grabar los datos de una cantidad no determinada de vehículos*.
- c) Ingresar un número de licencia y borrar físicamente. Si el número de licencia no existe mostrar un mensaje indicando que "El número de licencia no fue encontrado".
- d) *Listar los datos* del archivo.

**CODIFICACIÓN:**

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/*----- Estructura -----*/
struct fecha
{
    int dia;
    int mes;
    int anio;
};

/*----- Estructura Jerárquica -----*/
struct RegTaxis
{
    int nroLicencia;
    char ApeNomb[40];
    fecha fec; //Miembro "fec" de tipo estructura fecha.
};

/*----- Prototipos Auxiliares -----*/
int menuPrincipal();
void mensaje(char const *cadena);

/*----- Prototipos que dan solución al enunciado -----*/
void registrarTaxis(FILE *taxis); //PUNTO 01.
bool borradoTaxis(FILE *taxis); //PUNTO 02.
void listarTaxis(FILE *taxis); //PUNTO 03.
```

```

/*----- Bloque Principal -----*/
main()
{
    int N=0, nroLic=0, nOp = 0;
    int nroOp = 0;

    /*----- Abrir y verificar apertura del archivo -----*/
    FILE *ArchTaxis;           //Se abre de lectura para conservar
    ArchTaxis = fopen("taxis.dat", "w+b"); //los anteriores registros.
    if(ArchTaxis == NULL)
    { // Evalua, Si hubo error, muestra mensaje y termina.
        system("CLS");
        mensaje("Ocurrio un error en la apertura del Archivo...");
        exit(1);
    }
    do
    {
        nOp = menuPrincipal(); //Llama la función que muestra el menú.
        switch(nOp)
        {
            case 1:
                registrarTaxis(ArchTaxis);
                break;
            case 2:
                bool band;
                band = borradoTaxis(ArchTaxis);
                if(band)
                    ArchTaxis = fopen("taxis.dat", "r+b"); //Abro de nuevo el archivo, porque
                                                            //Fue cerrado en el borrado físico.
                break;
            case 3:
                listarTaxis(ArchTaxis);
                break;
            case 4:
                system("CLS");
                mensaje("F i n   d e l   P r o g r a m a");
                break;
            default:
                system("CLS");
                mensaje("Ha ingresado un numero no valido");
                break;
        } //Fin del switch().
    } while(nOp != 4); //Fin del Ciclo Do
} //Final del main().

```

```

void registrarTaxis(FILE *Taxis)
{
    RegTaxis reg; //crea la variable "reg" de tipo RegTaxis.
    int nroReg=0; //Variable contador del nro de registros almacenado.
    char continua='N';
    do{
        system("CLS");
        printf("\t*****\n");
        printf("\t** Registro Nro %3.0d **\n", ++nroReg);
        printf("\t*****\n\n");
        printf("\n\t Nro Licencia : "); scanf("%d",&reg.nroLicencia);
        _flushall();
        printf("\n\t Apellido y Nombre: "); gets(reg.ApeNomb);
        printf("\n\t Fecha Incorporacion: \n");
        printf("\n\t Dia: "); scanf("%d",&reg.fec.dia);
        printf("\n\t Mes: "); scanf("%d",&reg.fec.mes);
        printf("\n\t Añio: "); scanf("%d",&reg.fec.anio);
        fwrite(&reg, sizeof(RegTaxis), 1, Taxis); //Graba el registro lógico.
    } while(continua != 'N');
}

```

Registrar datos en miembro o campo de tipo struct

```

    _flushall();
    printf("\n\nContinua Registrando Taxis (S/N): ");
    scanf("%c", &continua);
}while(continua == 'S' || continua == 's');
system("CLS");
mensaje("F i n    de la    c a r g a"); //Muestra el mensaje.
} //Fin de la función registrarTaxis.

```

```

bool borradoTaxis(FILE *Taxis){
    int nroLicencia;
    bool bandEncontrado; //Variable para indicar si fue borrado o no.
    bandEncontrado = false; //Se inicializa en falso.
    RegTaxis reg;

    /*----- Se crea un archivo Temporal -----*/
    FILE *ArchTemp;
    ArchTemp = fopen("taxisTemp.dat", "w+b"); //Se crea un archivo de respaldo.

    /*----- Se muestra un título ----*/
    system("CLS");
    printf("\n\t*****");
    printf("\n\t**");
    printf("\n\t**          B O R R A D O   F I S I C O          **");
    printf("\n\t**");
    printf("\n\t*****");

    /*----- Se pide el ingreso del nro de licencia a buscar ----*/
    printf("\n\n\t Ingrese Nro de Licencia: ");
    scanf("%d", &nroLicencia);
    if (nroLicencia > 0 ){ //Si es mayor a cero se busca, en otro caso sale.
        rewind(Taxis); //Ubica el puntero en el primer registro del archivo.
        fread(&reg, sizeof(reg), 1, Taxis); //Lee el primer registro.
        while(!feof(Taxis)){ //Repetir hasta el último registro.
            if(nroLicencia != reg.nroLicencia){ //Si son diferente, Se guarda en el
                fwrite(&reg, sizeof(reg), 1, ArchTemp); //archivo temporal el registro leído.
            }else{ //si son iguales
                bandEncontrado = true; //Pone en verdadero indicando que fue encontrado y
            } // no lo guarda
            fread(&reg, sizeof(RegTaxis), 1, Taxis); //Continua leyendo.
        }
    }else{
        fclose(ArchTemp); //Cierra el archivo temporal.
    }
    if(bandEncontrado){
        fclose(ArchTemp); //Cierro archivo Temporal.
        fclose(Taxis); //Cierro archivo Maestro.
        remove("taxis.dat"); //Borro el archivo Maestro.
        rename("taxisTemp.dat", "taxis.dat"); //Cambio el nombre del Archivo Temporal.
        //Por el nombre del maestro (taxis.dat).
        mensaje("Nro de Licencia Encontrada, ha sido eliminado físicamente");
    }else{
        mensaje("Nro de Licencia No Encontrada, No se realizo el borrado");
        fclose(ArchTemp); //Cierro archivo Temporal.
    }
    system("CLS");
    mensaje("F I N    D E L   B O R R A D O ");
    remove("taxisTemp.dat"); //Elimina el archivo temporal creado.
    return bandEncontrado;
} //Fin de la función borradoTaxis.

```

```

void mensaje(char const *cadena)
{
    printf("\n\n\n*****\n");
    printf("\n %50s ", cadena );
    printf("\n\n*****\n\n\t");
    system("PAUSE");
} //Fin de la función mensaje().

```

```

void listarTaxis(FILE *Taxis){
    RegTaxis      reg; //Registro logico.

    /*----- Titulo del listado -----*/
    system("CLS");
    printf("\t*****\n");
    printf("\t**  L I S T A D O  D E  T A X I S  R E T I R A D O S  **\n");
    printf("\t*****\n\n");
    printf("\t*  Licencia \t Chofer\t \t Fecha      **\n\n");
    printf("\t*****\n\n");

    /*----- Listado de los Taxis -----*/
    rewind(Taxis); //Ubica el puntero en el primer registro del archivo.
    fread(&reg, sizeof(reg), 1, Taxis); //Leer el primer registro.
    if (feof(Taxis)){
        system("CLS");
        mensaje("El Archivo esta vacio\n No se Registro Informacion.");
    }else{
        while(!feof(Taxis)){ //Repite hasta el último registro.
            printf("\t %d      ", reg.nroLicencia);
            printf("\t %s      ", reg.ApeNomb );
            printf("\t %d/%d/%d      ", reg.fec.dia, reg.fec.mes, reg.fec.anio);
            printf("\n");
            fread(&reg, sizeof(RegTaxis), 1, Taxis); //Continua leyendo.
        }
    }
    mensaje("F i n   d e l   L i s t a d o");
} //Fin de la función listarTaxis.

```

```

int menuPrincipal()
{
    int opc=0;
    system("CLS");
    printf("\t*****\n");
    printf("\t**      M E N U      P R I N C I P A L      **\n");
    printf("\t*****\n");
    printf("\t**      **\n");
    printf("\t**      1-> Registrar      **\n");
    printf("\t**      **\n");
    printf("\t**      2-> Borrar      **\n");
    printf("\t**      **\n");
    printf("\t*****\n");
    printf("\t**      **\n");
    printf("\t**      3-> Listar Choferes      **\n");
    printf("\t**      **\n");
    printf("\t**      4-> S A L I R   d e l   S I S T E M A      **\n");
    printf("\t**      **\n");
    printf("\t*****\n\n");
    printf("  Seleccione Opcion: ");
    scanf("%d", &opc);
    return opc; //retorna el número de opción seleccionada.
} //Fin de la función menuPrincipal.

```