

Sistema de Gestión Hotelera en C++ almacenando en un .DAT

ESTRUCTURA DE DATOS. 2005

Muy a menudo necesitamos almacenar cierta cantidad de datos de forma más o menos permanente. La memoria del ordenador es volátil, y lo que es peor, escasa y cara. De modo que cuando tenemos que guardar nuestros datos durante cierto tiempo tenemos que recurrir a sistemas de almacenamiento más económicos, aunque sea a costa de que sean más lentos.

Durante la historia de los ordenadores se han usado varios métodos distintos para el almacenamiento de datos. Al principio se recurrió a cintas de papel perforadas, después a tarjetas perforadas. A continuación se pasó al soporte magnético, empezando por grandes rollos de cintas magnéticas abiertas.

Hasta aquí, todos los sistemas de almacenamiento externo eran secuenciales, es decir, no permitían acceder al punto exacto donde se guardaba la información sin antes haber partido desde el principio y sin haber leído toda la información, hasta el punto donde se encontraba la que estábamos buscando.

Con las cintas magnéticas empezó lo que con el tiempo sería el acceso aleatorio a los datos. Se podía reservar parte de la cinta para guardar cierta información sobre la situación de los datos, y añadir ciertas marcas que hicieran más sencillo localizarla.

Pero no fué hasta la aparición de los discos magnéticos cuando ésta técnica llegó a su sentido más amplio. En los discos es más sencillo acceder a cualquier punto de la superficie en poco tiempo, ya que se accede al punto de lectura y escritura usando dos coordenadas físicas. Por una parte la cabeza de lectura/escritura se puede mover en el sentido del radio del disco, y por otra el disco gira permanentemente, con lo que cualquier punto del disco pasa por la cabeza en un tiempo relativamente corto. Esto no pasa con las cintas, donde sólo hay una coordenada física.

Con la invención y proliferación de los discos se desarrollaron los ficheros de acceso aleatorio, que permiten acceder a cualquier dato almacenado en un fichero en relativamente poco tiempo.

Actualmente, los discos duros tienen una enorme capacidad y son muy rápidos, aunque aún siguen siendo lentos, en comparación con las memorias RAM. El caso de los CD es algo intermedio. En realidad son secuenciales en cuanto al modo de guardar los datos, cada disco sólo tiene una pista de datos grabada en espiral. Sin embargo, este sistema, combinado con algo de memoria RAM, proporciona un acceso muy próximo al de los discos duros.

En cuanto al tipo de acceso, en C y C++ podemos clasificar los archivos según varias categorías:

1. Dependiendo de la dirección del flujo de datos:

- a. De entrada: los datos se leen por el programa desde el archivo.
- b. De salida: los datos se escriben por el programa hacia el archivo.
- c. De entrada/salida: los datos pueden ser escritos o leídos.

2. Dependiendo del tipo de valores permitidos a cada byte:

- a. De texto: sólo están permitidos ciertos rangos de valores para cada byte. Algunos bytes tienen un significado especial, por ejemplo, el valor hexadecimal 0x1A marca el fin de fichero. Si abrimos un archivo en

modo texto, no será posible leer más allá de un byte con ese valor, aunque el fichero sea más largo.

- b. Binarios: están permitidos todos los valores para cada byte. En estos archivos el final del fichero se detecta de otro modo, dependiendo del soporte y del sistema operativo. La mayoría de las veces se hace guardando la longitud del fichero. Cuando queramos almacenar valores enteros, o en coma flotante, o imágenes, etc, deberemos usar este tipo de archivos.

3. Según el tipo de acceso:

- a. Archivos secuenciales: imitan el modo de acceso de los antiguos ficheros secuenciales almacenados en cintas magnéticas y
- b. Archivos de acceso aleatorio: permiten acceder a cualquier punto de ellos para realizar lecturas y/o escrituras.

4. Según la longitud de registro:

- a. Longitud variable: en realidad, en este tipo de archivos no tiene sentido hablar de longitud de registro, podemos considerar cada byte como un registro. También puede suceder que nuestra aplicación conozca el tipo y longitud de cada dato almacenado en el archivo, y lea o escriba los bytes necesarios en cada ocasión. Otro caso es cuando se usa una marca para el final de registro, por ejemplo, en ficheros de texto se usa el carácter de retorno de línea para eso. En estos casos cada registro es de longitud diferente.
- b. Longitud constante: en estos archivos los datos se almacenan en forma de registro de tamaño constante. En C usaremos estructuras para definir los registros. C dispone de funciones de librería adecuadas para manejar este tipo de ficheros.
- c. Mixtos: en ocasiones pueden crearse archivos que combinen los dos tipos de registros, por ejemplo, dBASE usa registros de longitud constante, pero añade un registro especial de cabecera al principio para definir, entre otras cosas, el tamaño y el tipo de los registros.

Es posible crear archivos combinando cada una de estas categorías, por ejemplo: archivos secuenciales de texto de longitud de registro variable, que son los típicos archivos de texto. Archivos de acceso aleatorio binarios de longitud de registro constante, normalmente usados en bases de datos. Y también cualquier combinación menos corriente, como archivos secuenciales binarios de longitud de registro constante, etc.

En cuanto a cómo se definen estas propiedades, hay dos casos. Si son binarios o de texto o de entrada, salida o entrada/salida, se define al abrir el fichero, mediante la función `fopen` en C o mediante el método `open` de `fstream` en C++.

En C++ es algo diferente, el constructor de las clases `ifstream`, `ofstream` y `fstream` admite los parámetros para abrir el fichero directamente, y también disponemos del método `open`, para poder crear el stream sin asociarlo con un fichero concreto y hacer esa asociación más tarde.

ENUNCIADO DEL PROGRAMA

Hotel Carlanchin necesita un programa que almacena los datos de cada huesped:

- | | |
|----------------|--------------------|
| - N° de cedula | - N° de habitación |
| - Nombre | - Tipo |
| - Apellido | - Valor |
| - Procedencia | - Facturación |
| - Teléfono. | - Consumo. |

En un archivo *.dat.

Diseñe un programa que nos permita realizar búsquedas por cliente, búsquedas por habitación e imprima facturación por cliente.

DATOS DEL PROGRAMA

Nombre del programa: Hotel Carlanchin
Nombre del archivo fuente: ARCHOTEL.CPP
Nombre del archivo ejecutable: ARCHOTEL.EXE
Fecha de creación: 3-06-2005
Compilador usado: TURBO C++ IDE 3.0

CODIGO FUENTE

```
#include <iomanip.h>
#include <process.h>
#include <fstream.h>
#include <conio.h>
#include <io.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <ctype.h>

#define hsenc 80000
#define hlujo 100000
#define false 1
#define true 0

char *NombreArchivo = "HOTEL.DAT";

typedef struct
{
    char cedula[10];
    char nombre[30];
    char apellido[30];
    char telefono[10];
    char procedencia[30];
    char estado;
    int NoHabitacion;
    long Nodias;
    int tipo;
    long valorH;
    long valorT;
    long consumo;
}Hotel_Carlanchin;

Hotel_Carlanchin Cliente;
int NHab;
```

```
int RevisaHab(int NHab); //REVISAR SI LA HABITACION ESTA OCUPADA
void Capturar(); //CAPTURA Y GUARDA LOS DATOS DE LOS HUESPEDES
void InConsumoXNumh(); //INSERTA CONSUMO DEL HUESPED X N° HABITACION
void InConsumoXCed(); //INSERTA CONSUMO DEL HUESPED X N° CEDULA
void Facturas(); //MUESTRA TODAS LAS FACTURAS
void Busqxcedula(); //BUSQUEDA DE HUESPED X N° CEDULA
void Busqxhabitacion(); //BUSQUEDA DE HUESPED X N° HABITACION
void Busqxapellido(); //BUSQUEDA DE HUESPED X APELLIDO & NOMBRE
void SalidaHuesped(); //DA SALIDA AL HUESPED Y LIBERA HABITACION
void MostrarHusp(); //MUESTRA LISTA DE HUESPEDES ACTUALES Y SALIENTES
void reset(); // RESETEA EL ARCHIVO HOTEL.DAT

void main()
{
    int opcion;
    char op;
    while(op != 27)
    {
        clrscr();
        cout<<"\n\tEiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii>>" ;
        cout<<"\n\t0" ;
        cout<<"\n\t0 -- HOTEL CARLANCHIN -- " ;
        cout<<"\n\t0" ;
        cout<<"\n\tEiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii1" ;
        cout<<"\n\t0" ;
        cout<<"\n\t0 1.- INGRESAR DATOS " ;
        cout<<"\n\t0 2.- INGRESAR CONSUMO DEL HUESPED " ;
        cout<<"\n\t0 3.- MOSTRAR FACTURAS " ;
        cout<<"\n\t0 4.- BUSQUEDA DE HUESPEDES " ;
        cout<<"\n\t0 5.- MOSTRAR LISTA DE HUESPEDES " ;
        cout<<"\n\t0 6.- SALIDA DE HUESPEDES " ;
        cout<<"\n\t0 7.- ELIMINAR REGISTROS " ;
        cout<<"\n\t0 ESC.- SALIR " ;
        cout<<"\n\tEiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii4" ;
        cout<<"\n\n\t--> DIGITE UNA OPCION : ";
        op=getch();
        switch (op)
        {
            case '1': Capturar();
                        break;
            case '2': clrscr();
                        cout<<"\n\n --- INGRESAR CONSUMO HUESPED ---";
                        cout<<"\n\n\t 1. N° CEDULA";
                        cout<<"\n\t 2. N° HABITACION";
                        cout<<"\n\n\t DIGITE LA OPCION:";
                        cin>>opcion;
                        if(opcion==1){InConsumoXCed();}
                        if(opcion==2){InConsumoXNumh();}
                        break;
            case '3': Facturas();
                        break;
            case '4': clrscr();
                        cout<<"\n\n --- BUSQUEDA DE HUESPEDES ---";
                        cout<<"\n\n\t 1. N° HABITACION";
                        cout<<"\n\t 2. APELLIDO - NOMBRE";
                        cout<<"\n\t 3. N° CEDULA";
                        cout<<"\n\n\t DIGITE LA OPCION:";
                        cin>>opcion;
                        if(opcion==1){Busqxhabitacion();}
                        if(opcion==2){Busqxapellido();}
                        if(opcion==3){Busqxcedula();}
                        break;
            case '5': MostrarHusp();
                        break;
            case '6': SalidaHuesped();
                        break;
            case '7': reset();
                        break;
            default : if(op == 27){exit(0);}
        }
    }
}

//*****
void Capturar()
{
    int sw2;
    char res, res1;

    do{
        clrscr();
        cout<<"\ndIGITE EL N° DE HABITACION: ";
        cin>>NHab;
        sw2=RevisaHab(NHab);
    }while(sw2==false);
    ofstream Archivo;
```

```

Archivo.open(NombreArchivo, ios::ate);
if (!Archivo)
{
    cout << "Error en la apertura del archivo \a";
}
else
{
    Cliente.NoHabitacion=NHab;
    cout<<"\nNombre: ";
    gets(Cliente.nombre);
    cout<<"Apellido: ";
    gets(Cliente.apellido);
    cout<<"Cedula: ";
    gets(Cliente.cedula);
    cout<<"Telefono: ";
    gets(Cliente.telefono);
    cout<<"Procedencia: ";
    gets(Cliente.procedencia);
    cout<<"Tipo de Habitacion : ";
    cout<<" 1.H. Sencilla";
    cout<<" 2.H. Lujo : ->";
    cin>>Cliente.tipo;
    if(Cliente.tipo==1){Cliente.valorH=hsenc;}
    if(Cliente.tipo==2){Cliente.valorH=hlujo;}
    cout<<"Nº Dias: ";
    cin>>Cliente.Nodias;
    Cliente.estado='H';
    Cliente.consumo=0;
    Cliente.valorT=(Cliente.valorH*Cliente.Nodias);
    cout<<"\nDesea Guardar Registro S/N";
    res=getch();
    if(res=='s' || res=='S')
    {
        Archivo.write((char *)&Cliente, sizeof(Cliente));
    }
}
Archivo.close();
getch();
}

/*****/
int RevisaHab(int NHab)
{
    int sw=true;

    fstream rd(NombreArchivo, ios::in|ios::out|ios::binary);
    do
    {
        rd.read((char *)&Cliente, sizeof(Cliente));
        if(rd.eof())
            break;
        if(NHab == Cliente.NoHabitacion)
        {
            sw=false;
            cout<<"\n-----";
            cout<<"\n--HABITACION OCUPADA--";
            cout<<"\n-----";
            getch();
            break;
        }
    }while(!rd.eof());
    rd.close();
    return sw;
}

/*****/
void Facturas()
{
    int sw,numReg=0;
    fstream Archivo;
    clrscr();
    Archivo.open(NombreArchivo,ios::in|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                break;
            }
            cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);

```

```

        clrscr();
        cout<<"\n/-----/";
        cout<<"\n FACTURA N° : "<<numReg;
        if(Cliente.estado=='N'){cout<<" CANCELADA";}
        cout<<"\n/-----/";
        cout<<"\n Estado : "<<Cliente.estado;
        cout<<"\n Apellido : "<<Cliente.apellido;
        cout<<"\n Nombre : "<<Cliente.nombre;
        cout<<"\n Cedula : "<<Cliente.cedula;
        cout<<"\n N° Habitación : ";
        if(Cliente.NoHabitacion==0){cout<<"---";}
        else{cout<<Cliente.NoHabitacion;}
        if(Cliente.tipo==1){cout<<"\n Tipo : Habitacion Sencilla";}
        if(Cliente.tipo==2){cout<<"\n Tipo : Habitacion de Lujo";}
        cout<<"\n Valor Hab. Dia : "<<Cliente.valorH;
        cout<<"\n N°Dias : "<<Cliente.Nodias;
        cout<<"\n Valor Consumo : $ "<<Cliente.consumo;
        cout<<"\n /-----/";
        cout<<"\n Valor Total : $ "<<Cliente.valorT;
        cout<<"\n/-----/";
        cout<<"\n Estado: (H)Hospedado---(N)No Hospedado ";
        cout<<"\n/-----/\n";
        getch();
    }while(!Archivo.eof());
    Archivo.close();
}

getch();
}

/*****
void SalidaHuesped()
{
    long Pconsumo;
    int numReg=0, Numh;
    clrscr();
    fstream Archivo;
    Archivo.open(NombreArchivo, ios::in|ios::out|ios::binary);

    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        cout<<"\n ----- SALIDA DE HUESPED ----- ";
        cout<<"\n\n\tDIGITE N° HABITACION: ";
        cin>>Numh;
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                clrscr();
                cout<<"\n\n\n\n\tRegistro no encontrado...";
                break;
            }
            if(Numh==Cliente.NoHabitacion)
            {
                cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
                cout<<"\n/----- SALIDA DE HUESPED -----/";
                cout<<"\n/-----/";
                cout<<"\n FACTURA N° : "<<numReg;
                cout<<"\n/-----/";
                cout<<"\n Apellido : "<<Cliente.apellido;
                cout<<"\n Nombre : "<<Cliente.nombre;
                cout<<"\n Cedula : "<<Cliente.cedula;
                cout<<"\n N° Habitación : "<<Cliente.NoHabitacion;
                if(Cliente.tipo==1){cout<<"\n Tipo : Habitacion Sencilla";}
                if(Cliente.tipo==2){cout<<"\n Tipo : Habitacion de Lujo";}
                cout<<"\n Valor Hab. Dia : "<<Cliente.valorH;
                cout<<"\n N°Dias : "<<Cliente.Nodias;
                cout<<"\n Valor Consumo : $ "<<Cliente.consumo;
                cout<<"\n /-----/";
                cout<<"\n Valor Total : $ "<<Cliente.valorT;
                cout<<"\n/-----/\n";
                getch();
                Cliente.NoHabitacion=0;
                Cliente.estado='N';
                Archivo.seekp((numReg-1)*sizeof(Cliente));
                Archivo.write((char *)&Cliente, sizeof(Cliente));
                break;
            }
        }while(!Archivo.eof());
        Archivo.close();
    }
}

```

```

}
/*****
void InConsumoXCed()
{
    char ced[10];
    long Pconsumo;
    int numReg=0;
    clrscr();
    fstream Archivo;
    Archivo.open(NombreArchivo,ios::in|ios::out|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        cout<<"\n ----- INGRESAR CONSUMO DE HUESPED ----- ";
        cout<<"\n\n\tDigite N$ Cedula:";
        gets(ced);
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                clrscr();
                cout<<"\n\n\n\n\tRegistro no encontrado...";
                break;
            }
            if(strcmp(ced,Cliente.cedula)==0)
            {
                if(Cliente.estado=='N')
                {
                    cout<<"\n-----";
                    cout<<"\n CLIENTE NO HOSPEDADO ";
                    cout<<"\n-----";
                    break;
                }
                else
                {
                    cout<<"\n ----- INGRESAR CONSUMO DE HUESPED ----- ";
                    cout<<"\n N$ Habitacion      : "<<Cliente.NoHabitacion;
                    cout<<"\n Apellido        : "<<Cliente.apellido;
                    cout<<"\n Nombre          : "<<Cliente.nombre;
                    cout<<"\n Valor Consumido   : $ "<<Cliente.consumo;
                    cout<<"\n Valor Total     : $ "<<Cliente.valorT;
                    cout<<"\n/-----/\n";
                    cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
                    cout<<"\n DIGITE EL CONSUMO : $ ";
                    cin>>Pconsumo;
                    Cliente.consumo=Cliente.consumo+Pconsumo;
                    Cliente.valorT=Cliente.valorT+Pconsumo;
                    Archivo.seekp((numReg-1)*sizeof(Cliente));
                    Archivo.write((char *)&Cliente,sizeof(Cliente));
                    break;
                }
            }
        }while(!Archivo.eof());
        Archivo.close();
    }
    getch();
}
/*****
void InConsumoXNumh()
{
    long Pconsumo;
    int numReg=0,Numh;
    clrscr();
    fstream Archivo;
    Archivo.open(NombreArchivo,ios::in|ios::out|ios::binary);

    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        cout<<"\n ----- INGRESAR CONSUMO DE HUESPED ----- ";
        cout<<"\n\n\tDigite N$ de Habitaci n: ";
        cin>>Numh;
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())

```

```

        {
            clrscr();
            cout<<"\n\n\n\n\n\tRegistro no encontrado...";
            break;
        }
        if(Numh==Cliente.NoHabitacion)
        {
            cout<<"\n ----- INGRESAR CONSUMO DE HUESPED ----- ";
            cout<<"\n N$ Habitacion      : "<<Cliente.NoHabitacion;
            cout<<"\n Apellido          : "<<Cliente.apellido;
            cout<<"\n Nombre            : "<<Cliente.nombre;
            cout<<"\n Valor Consumido    : $ "<<Cliente.consumo;
            cout<<"\n Valor Total      : $ "<<Cliente.valorT;
            cout<<"\n/-----/\n";
            cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
            cout<<"\n DIGITE EL CONSUMO: $ ";
            cin>>Pconsumo;
            Cliente.consumo=Cliente.consumo+Pconsumo;
            Cliente.valorT=Cliente.valorT+Pconsumo;
            Archivo.seekp((numReg-1)*sizeof(Cliente));
            Archivo.write((char *)&Cliente,sizeof(Cliente));
            break;
        }
    }while(!Archivo.eof());
    Archivo.close();
}
getch();
}
/*****
void BusqxApellido()
{
    char ape[30],nom[50];
    int numReg;
    fstream Archivo;
    Archivo.open(NombreArchivo,ios::in|ios::out|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        numReg=0;
        clrscr();
        cout<<"\n/----- BUSQUEDA DE HUESPED -----/";
        cout<<"\n\n\tDigite el Apellido: ";
        gets(ape);
        cout<<"\n\tDigite el Nombre: ";
        gets(nom);
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                clrscr();
                cout<<"\n\n\n\n\n\tRegistro no encontrado...";
                break;
            }
            if((strcmp(ape,Cliente.apellido)==0)&&(strcmp(nom,Cliente.nombre)==0))
            {
                if(Cliente.estado=='N')
                {
                    cout<<"\n-----";
                    cout<<"\n CLIENTE NO HOSPEDADO ";
                    cout<<"\n-----";
                    break;
                }
                else
                {
                    cout<<"\n/----- HUESPED ENCONTRADO -----/\n";
                    cout<<"\n N$ Habitacion      : "<<Cliente.NoHabitacion;
                    cout<<"\n Apellido          : "<<Cliente.apellido;
                    cout<<"\n Nombre            : "<<Cliente.nombre;
                    cout<<"\n/-----/\n";
                    getch();
                    cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
                    break;
                }
            }
        }while(!Archivo.eof());
        Archivo.close();
    }
    getch();
}
/*****/
void Busqxcedula()

```



```

{
    char ced[10];
    int numReg=0;
    clrscr();
    fstream Archivo;
    Archivo.open(NombreArchivo,ios::in|ios::out|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        cout<<"\n/----- BUSQUEDA DE HUESPED -----/";
        cout<<"\n\n\tDigite N° Cedula: ";
        gets(ced);
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                clrscr();
                cout<<"\n\n\n\n\n\tRegistro no encontrado...";
                break;
            }

            if(strcmp(ced,Cliente.cedula)==0)
            {
                if(Cliente.estado=='N')
                {
                    cout<<"\n-----";
                    cout<<"\n CLIENTE NO HOSPEDADO ";
                    cout<<"\n-----";
                    break;
                }
                else
                {
                    cout<<"\n/----- HUESPED ENCONTRADO -----/\n";
                    cout<<"\n N° Habitación : "<<Cliente.NoHabitacion;
                    cout<<"\n Apellido : "<<Cliente.apellido;
                    cout<<"\n Nombre : "<<Cliente.nombre;
                    cout<<"\n Cedula : "<<Cliente.cedula;
                    cout<<"\n N°Días : "<<Cliente.Nodias;
                    cout<<"\n Valor Habitación : "<<Cliente.valorH;
                    cout<<"\n Valor Consumo : $ "<<Cliente.consumo;
                    cout<<"\n Valor Total : $ "<<Cliente.valorT;
                    cout<<"\n/-----/\n";
                    getch();
                    cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
                    break;
                }
            }
        }while(!Archivo.eof());
        Archivo.close();
    }

    getch();
}
/*****

```

```

void Busqxhabitacion()
{
    int Numh;
    int numReg;
    fstream Archivo;
    Archivo.open(NombreArchivo,ios::in|ios::out|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
        getch();
    }
    else
    {
        numReg=0;
        clrscr();
        cout<<"\n/----- BUSQUEDA DE HUESPED -----/";
        cout<<"\n\n\tDigite el N° de habitación: ";
        cin>>Numh;
        clrscr();
        do
        {
            numReg++;
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                clrscr();
            }
        }
    }
}

```

```

        cout<<"\n\n\n\n\n\tRegistro no encontrado...";
        break;
    }
    if(Numh==Cliente.NoHabitacion)
    {
        cout<<"\n/----- HUESPED ENCONTRADO -----/\n";
        cout<<"\n N§ Habitación      : "<<Cliente.NoHabitacion;
        cout<<"\n Apellido        : "<<Cliente.apellido;
        cout<<"\n Nombre          : "<<Cliente.nombre;
        cout<<"\n Cedula          : "<<Cliente.cedula;
        cout<<"\n N§Días            : "<<Cliente.Nodias;
        cout<<"\n Valor Habitación : "<<Cliente.valorH;
        cout<<"\n Valor Consumo   : $ "<<Cliente.consumo;
        cout<<"\n Valor Total    : $ "<<Cliente.valorT;
        cout<<"\n/-----/\n";
        getch();
        cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
        break;
    }
    }while(!Archivo.eof());
    Archivo.close();
}

getch();
}

/*****
void MostrarHusp()
{
    int sw;
    fstream Archivo;
    clrscr();
    Archivo.open(NombreArchivo,ios::in|ios::binary);
    if (!Archivo)
    {
        cout << "Error en la apertura del archivo \a";
    }
    else
    {
        cout<<"\n\n\n/-----/"
        cout<<"\nN§ Hab.  Apellido  Nombre  Cedula  Procedencia  Telefono  Estado
\n\n";
        do
        {
            Archivo.read( (char *)&Cliente, sizeof(Cliente) );
            if(Archivo.eof())
            {
                break;
            }

            if(Cliente.NoHabitacion==0){cout<<setiosflags(ios::left)<<setw(9)<<"---";}
            else{cout<<setiosflags(ios::left)<<setw(9)<<Cliente.NoHabitacion;}
            cout<<setw(11)<<Cliente.apellido<<setprecision(2);
            cout<<setw(11)<<Cliente.nombre<<setprecision(2);
            cout<<setw(10)<<Cliente.cedula<<setprecision(2);
            cout<<setw(13)<<Cliente.procedencia<<setprecision(2);
            cout<<setw(12)<<Cliente.telefono<<setprecision(2);
            cout<<setw(9)<<Cliente.estado<<setprecision(2)<<'\\n';

            }while(!Archivo.eof());
            cout<<"\n/-----/"
            cout<<"\n Estado:(H)Hospedado----(N)No Hospedado ";
            cout<<"\n/-----/\n";
            Archivo.close();
        }
        getch();
    }
}
*****/

void reset()
{
    ofstream arch;

    arch.open ("HOTEL.DAT", ios::out);
    arch.close();

    cout<<"\n\t/-----/"
    cout <<"\n\t El archivo esta reseteado!!!";
    cout<<"\n\t/-----/"
    getch();
}

```

IMPRESIÓN DE PANTALLA DE RESULTADO

MENU

```
-- HOTEL CARLANCHIN --

1.- INGRESAR DATOS
2.- INGRESAR CONSUMO DEL HUESPED
3.- MOSTRAR FACTURAS
4.- BUSQUEDA DE HUESPEDES
5.- MOSTRAR LISTA DE HUESPEDES
6.- SALIDA DE HUESPEDES
7.- ELIMINAR REGISTROS
ESC.- SALIR

--> DIGITE UNA OPCION :
```

INGRESAR CONSUMO DEL HUESPED

```
----- INGRESAR CONSUMO DE HUESPED -----
Nº Habitación      : 101
Apellido           : PEREZ
Nombre             : CARLOS
Valor Consumido    : $ 0
Valor Total        : $ 400000
-----/
DIGITE EL CONSUMO: $ 550_
```

MOSTRAR FACTURAS

```
FACTURA Nº       : 2
-----/
Estado            : H
Apellido          : PETRO
Nombre            : MARIA
Cedula            : 98556542
Nº Habitación     : 201
Tipo              : Habitación de Lujo
Valor Hab. Dia    : 100000
NºDias            : 4
Valor Consumo     : $ 0
-----/
Valor Total       : $ 400000
-----/
Estado:<H>Hospedado----<N>No Hospedado
-----/
```

BUSQUEDA DE HUESPED

```

/----- HUESPED ENCONTRADO -----/
Nº Habitación      : 105
Apellido           : RUIZ
Nombre            : SANDRA
Cedula            : 1236595
NºDias            : 5
Valor Habitación   : 100000
Valor Consumo      : $ 15000
Valor Total        : $ 515000
/-----/

```

SALIDA DE HUESPED

```

/----- SALIDA DE HUESPED -----/
FACTURA Nº        : 1
/-----/
Apellido           : PEREZ
Nombre            : CARLOS
Cedula            : 7895631
Nº Habitación      : 101
Tipo              : Habitación de Lujo
Valor Hab. Dia     : 100000
NºDias            : 4
Valor Consumo      : $ 550
/-----/
Valor Total        : $ 400550
/-----/

```

LISTA DE HUESPEDES

```

/-----/
Nº Hab.  Apellido  Nombre  Cedula  Procedencia  Telefono  Estado
101      PEREZ     CARLOS  7895631  MEDELLIN    7826598   H
201      PETRO     MARIA   98556542 VALENCIA    785965    H
202      TALEO     CARLOS  4789565  BOGOTA      22355     H
105      RUIZ       SANDRA  1236595  CALI        25865     H
/-----/
Estado: <H>Hospedado----<N>No Hospedado
/-----/

```

