

**Software para Almacenamiento de Hojas de Vida**  
**Código Fuente**

**Versión 2.0**

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

## 9. CODIGO FUENTE

El Software HVIDA V.01 fue codificado en el Lenguaje C/C++, y compilado con Turbo C++ V.3.1.

```
Size: 35105
Last Modified: Tue Oct 25 21:12:36 2005
File Parse State: File unchanged
Classes: 0
Functions: 21
Lines: 1078
Blank Lines: 31
Code Lines: 992
Comment Lines: 56
Statements: 907
Ratio Comment/Code: 0.06
```

```
/*-----*/
/*
/*      NOMBRE: HVIDA.CPP - SOFTWARE ALMACENAMIENTO DE HOJAS DE VIDA
/*
/*-----*/

/*-----LIBRERIAS-----*/
#include <iomanip.h>
#include <fstream.h>
#include <conio.h>
#include <stdio.h>
#include <string.h>
#include <dos.h>
#include <ctype.h>
#include <stdlib.h>

/*-----DECLARACION DE CONSTANTES-----*/
#define PAUSA delay(2500);
#define FALSE 1
#define TRUE 0

/*-----ARCHIVOS EXTERNOS-----*/
char *ArchivoHV = "HVIDAS.DAT";//ARCHIVO DE HOJAS DE VIDA
char *ArchivoUSU = "HVUSUARIOS.DAT";//ARCHIVO DE USUARIOS

/*-----ESTRUCTURA DE DATOS HOJAS DE VIDA-----*/
typedef struct
{
    int    codigoHV;
    char   fechaing[30];
    /*-----CAMPOS DATOS PERSONALES*/
    char   cedula[10];
    char   apellido1[30];
    char   apellido2[30];
    char   nombre1[30];
    char   nombre2[30];
    char   telefono[10];
    char   direccion[30];
    char   fndia[5];
    char   fnmes[5];
    char   fnanio[5];
    char   lugarnac[30];
    char   cargo[30];
    int    estadociv;
    char   estado;
    /*-----CAMPOS DATOS ACADEMICOS*/
    int    tipobachi;
    char   fechabachi[30];
    char   colbachi[30];
    int    opcesupe;
    int    tipoesupe;
    char   esupe[30];
    char   fechaesupe[30];
    char   uesupe[30];
    int    opcpostr;
    char   postgrado[30];
    char   fechapost[30];
    char   upostgrado[30];
    int    opciidioma;
    char   idioma[100];
    int    opcsiste;
    char   sistemas[100];
}
```



Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

/*-----FUNCION MENU-----*/
void Menu(){
    char opc;
    int opc3;
    textattr(7);
    while(opc != 27){
        Marco2();
        cout<<"\n\t\t";
        cout<<"\n\t\t 1.- REGISTRAR HOJA DE VIDA";
        cout<<"\n\t\t 2.- CONSULTAR HOJA DE VIDA";
        cout<<"\n\t\t 3.- MOSTRAR CODIGOS EMPLEADOS";
        cout<<"\n\t\t 4.- MODIFICAR HOJA DE VIDA";
        cout<<"\n\t\t 5.- DESVINCULAR EMPLEADO";
        cout<<"\n\t\t 6.- ELIMINAR TODOS LOS REGISTROS";
        cout<<"\n\t\t 7.- REGISTRAR USUARIO";
        cout<<"\n\t\t [ESC] - SALIR";
        cout<<"\n\t\t";
        cout<<"\n\t\tQ";
        cout<<"\n\n\t\t --> DIGITE UNA OPCION : ";
        textattr(10);cprintf("[ ]\b\b");
        opc=getch();
        textattr(7);
        switch (opc){
            case '1': RegistrarHV();
                        break;
            case '2': do{
                            Marco2();
                            cout<<"\n\t\t";
                            cout<<"\n\t\t CONSULTAR HOJA DE VIDA POR:";
                            cout<<"\n\t\t 1.- CODIGO DE EMPLEADO";
                            cout<<"\n\t\t 2.- APELLIDOS - NOMBRE";
                            cout<<"\n\t\t 3.- N DE CEDULA";
                            cout<<"\n\t\t 0.- SALIR";
                            cout<<"\n\t\t";
                            cout<<"\n\n\t\t --> DIGITE UNA OPCION:";
                            textattr(10);cprintf("[ ]\b\b");
                            cin>>opc3;
                            textattr(7);
                        }while(opc3>=4);
                        if(opc3==0){break;}
                        if(opc3==1){BusqXCodigo();}
                        if(opc3==2){BusqXapellido();}
                        if(opc3==3){BusqXcedula();}
                        break;

            case '3': ListarHV();
                        break;
            case '4': Modificar();
                        break;
            case '5': Desvinemple();
                        break;
            case '6': Reset();
                        break;
            case '7': RegistrarU();
                        break;
            default : if(opc == 27){Marco7();}
        }
    }
}

/*-----FUNCION VALIDA LOGIN & PASSWORD USUARIO-----*/
int ValidaPassw()
{
    char pass[10],login[10];
    int numReg;
    int sw=FALSE;
    fstream Archivo;
    Archivo.open(ArchivoUSU,ios::in|ios::out|ios::binary);
    if (!Archivo){
        Marco3();
    }
    else{
        numReg=0;
        clrscr();
        Marcol();
        cout<<"\n\n\t\t ----- VALIDACION DE USUARIO -----";
        cout<<"\n\n\t\t DIGITE EL LOGIN      : ";
        gets(login);
        cout<<"\n\n\t\t DIGITE EL PASSWORD     : ";

```

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

gets(pass);
clrscr();
do{
    numReg++;
    Archivo.read( (char *)&usuario, sizeof(usuario) );
    if(Archivo.eof()){
        clrscr();
        Marco2();
        cout<<"\n\t\t";
        cout<<"\n\t\t"----- USUARIO NO VALIDO -----;
        cout<<"\n\t\t";
        cout<<"\n\t\t\u00A0";
        PAUSA;
        break;
    }
    //strcmp compara las cadenas sin distinguir Mayusculas
    if((strcmp(pass,usuario.password)==0)&&(strcmp(login,usuario.login)==0)){
        Marco1();
        cout<<"\n\n\n\t\t"----- BIENVENIDO -----";
        cout<<"\n\n\n\t\t"USUARIO : "<usuario.nombreU<<"
"<usuario.apellidoU;
        sw=TRUE;
        delay(3000);
        cout<<resetiosflags ios::left<<setiosflags(ios::fixed | ios::showpoint);
        break;
    }
}while(!Archivo.eof());
Archivo.close();
}
return sw;//Retorna False o True - Validar usuario
}

/*-----FUNCION CAPTURA & GUARDA LOS DATOS DE LAS HOJAS VIDA-----*/
void RegistrarHV()
{
    int sw=FALSE;
    char res;
    do{
        clrscr();
        Marco1();
        cout<<"\n\n\t\t DIGITE EL CODIGO DEL EMPLEADO [0]-SALIR: ";
        textattr(10);printf("[ ]\b\b\b\b");
        cin>>CodigoE;//Nuevo Codigo
        textattr(7);
        sw=RevisaCod(CodigoE);//Revisa si el codigo ya es utilizado
    }while(sw==FALSE);
    if (CodigoE != 0){
        ofstream Archivo;
        Archivo.open(ArchivoHV, ios::ate);
        if (!Archivo){
            Marco3();
        }
        else{
            empleado.codigoHV=CodigoE;
            empleado.estado='V';
            cout<<"\n\t\t REGISTRAR HOJA DE VIDA ";
            cout<<"\n\n\t\t FECHA DE VINCULACION (dd/mm/aa): ";
            gets(empleado.fechaing);
            cout<<"\n\t\t-----DATOS PERSONALES-----\n";
            cout<<"\n\t\tPrimer Apellido: ";
            gets(empleado.apellido1);
            cout<<"\t\tSegundo Apellido: ";
            gets(empleado.apellido2);
            cout<<"\t\tPrimer Nombre: ";
            gets(empleado.nombre1);
            cout<<"\t\tSegundo Nombre: ";
            gets(empleado.nombre2);
            cout<<"\t\tCedula: ";
            gets(empleado.cedula);
            cout<<"\t\tTelefono: ";
            gets(empleado.telefono);
            cout<<"\t\tDireccin: ";
            gets(empleado.direccion);
            cout<<"\t\tFecha de Nacimiento: ";
            do{
                cout<<"\n\t\tDigite el da (dd): ";
                gets(empleado.fndia);
            }while((atoi(empleado.fndia)>=32)|| (atoi(empleado.fndia)<=0));
            do{
                cout<<"\t\tDigite el mes (mm): ";

```

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

        gets(empleado.fnmes);
    }while((atoi(empleado.fnmes)>=13) || (atoi(empleado.fnmes)<=0));
    do{
        cout<<"\t\tDigite el ao (aaaa): ";
        gets(empleado.fnanio);
    }while((atoi(empleado.fnanio)<=1900) || (atoi(empleado.fnanio)>=2005));
    cout<<"\t\tLugar de Nacimiento: ";
    gets(empleado.lugarnac);
    cout<<"\t\tCargo Actual: ";
    gets(empleado.cargo);
    do{
        cout<<"\t\tEstado Civil : ";
        cout<<"\t 1.Soltero(a)";
        cout<<"\t 2.Casado (a): -->";
        textattr(10);cprintf("[ ]\b\b");
        cin>>empleado.estadociv;
        textattr(7);
    }while((empleado.estadociv<1) || (empleado.estadociv)>=3);
    cout<<"\n\t\t-----DATOS ACADEMICOS-----\n";
    do{
        cout<<"\n\t\t----BACHILLERATO----: ";
        cout<<"\n\t\t 1.Clsico";
        cout<<"\n\t\t 2.Comercial";
        cout<<"\n\t\t 3.Tcnico : -->";
        textattr(10);cprintf("[ ]\b\b");
        cin>>empleado.tipobachi;
        textattr(7);
    }while((empleado.tipobachi<1) || (empleado.tipobachi)>=4);
    cout<<"\n\t\tNombre de la Institucin: ";
    gets(empleado.colbachi);
    cout<<"\t\tAo Finalizacin: ";
    gets(empleado.fechabachi);
    do{
        cout<<"\n\t\t----EDUCACION SUPERIOR----: ";
        cout<<"\n\t\t 1.Si";
        cout<<"\n\t\t 2.No : -->";
        textattr(10);cprintf("[ ]\b\b");
        cin>>empleado.opcesupe;
        textattr(7);
    }while((empleado.opcesupe<1) || (empleado.opcesupe)>=3);
    if(empleado.opcesupe==1){
        do{
            cout<<"\n\t\t 1.Tnico";
            cout<<"\n\t\t 2.Tecnolgico";
            cout<<"\n\t\t 3.Profesional : -->";
            textattr(10);cprintf("[ ]\b\b");
            cin>>empleado.tipoesupe;
            textattr(7);
        }while((empleado.tipoesupe<1) || (empleado.tipoesupe)>=4);
        cout<<"\n\t\tTitulo Obtenido: ";
        gets(empleado.esupe);
        cout<<"\t\tNombre de la Institucin: ";
        gets(empleado.uesupe);
        cout<<"\t\tAo Finalizacin: ";
        gets(empleado.fechaesupe);
        do{
            cout<<"\n\t\t--POSTGRADO--: ";
            cout<<"\n\t\t 1.Si";
            cout<<"\n\t\t 2.No : -->";
            textattr(10);cprintf("[ ]\b\b");
            cin>>empleado.opcpst;
            textattr(7);
        }while((empleado.opcpst<1) || (empleado.opcpst)>=3);
        if(empleado.opcpst==1){
            cout<<"\n\t\tTitulo Obtenido: ";
            gets(empleado.postgrado);
            cout<<"\t\tNombre de la Institucin: ";
            gets(empleado.upostgrado);
            cout<<"\t\tAo Finalizacin: ";
            gets(empleado.fechapost);
        }
    }
    do{
        cout<<"\n\t\t----IDIOMAS----: ";
        cout<<"\n\t\t 1.Si";
        cout<<"\n\t\t 2.No : -->";
        textattr(10);cprintf("[ ]\b\b");
        cin>>empleado.opcidioma;
        textattr(7);
    }while((empleado.opcidioma<1) || (empleado.opcidioma)>=3);

```

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

        if(empleado.opcidioma==1){
            cout<<"\n\t\t Que idioma(s) conoce?: ";
            gets(empleado.idioma);
        }
        do{
            cout<<"\n\t\t ----SISTEMAS----: ";
            cout<<"\n\t\t 1.Si";
            cout<<"\n\t\t 2.No : -->";
            textattr(10);cprintf("[ ]\b\b");
            cin>>empleado.opcsiste;
            textattr(7);
        }while((empleado.opcsiste<1)|| (empleado.opcsiste)>=3);
        if(empleado.opcsiste==1){
            cout<<"\n\t\t Programa(s) que maneja?: ";
            gets(empleado.sistemas);
        }
        do{
            cout<<"\n\t\t -----DATOS LABORALES-----\n";
            cout<<"\n\t\t ----EXPERIENCIA LABORAL----: ";
            cout<<"\n\t\t 1.Si";
            cout<<"\n\t\t 2.No : -->";
            textattr(10);cprintf("[ ]\b\b");
            cin>>empleado.opclaboral;
            textattr(7);
        }while((empleado.opcsiste<1)|| (empleado.opcsiste)>=3);
        if (empleado.opclaboral==1){
            cout<<"\n\t\t Nombre de la Empresa: ";
            gets(empleado.empresa);
            cout<<"\t\t Cargo que Ocupaba: ";
            gets(empleado.cargoant);
            cout<<"\t\t Nombre del Jefe Inmediato: ";
            gets(empleado.jefe);
            cout<<"\t\t Telefono de la Empresa: ";
            gets(empleado.telempresa);
            cout<<"\t\t Fecha de Ingreso (dd/mm/aa): ";
            gets(empleado.fingreso);
            cout<<"\t\t Fecha de Retiro (dd/mm/aa): ";
            gets(empleado.fretiro);
        }
        cout<<"\n\t\t Desea Guardar Registro? [S]-GUARDAR";
        textattr(10);cprintf("[ ]\b\b");
        res=getch();
        textattr(7);
        if(res=='s' || res=='S'){
            Archivo.write((char *)&empleado, sizeof(empleado));
            Marco4();
        }
        else{
            Marco5();
        }
    }
    Archivo.close();
}

/*-----FUNCION VALIDA CODIGO A UTILIZAR-----*/
int RevisaCod(int CodigoE)
{
    int sw=TRUE;
    fstream rd(ArchivoHV, ios::in|ios::out|ios::binary);
    do{
        rd.read((char *)&empleado, sizeof(empleado));
        if(rd.eof())
            break;
        if(CodigoE == empleado.codigoHV){//Compara Codigo
            sw=FALSE;
            clrscr();
            Marco2();
            cout<<"\n\t\t ";
            cout<<"\n\t\t CODIGO DIGITADO EN USO!! ";
            cout<<"\n\t\t ";
            cout<<"\n\t\t ";
            PAUSA;
            break;
        }
    }while(!rd.eof());
    rd.close();
    return sw;
}

```





Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

        cout<<"\n\t\t Empresa          : "<<empleado.empresa;
        cout<<"\n\t\t Cargo            : "<<empleado.cargoant;
        cout<<"\n\t\t Jefe Inmediato     : "<<empleado.jefe;
        cout<<"\n\t\t Telefono          : "<<empleado.telempresa;
        cout<<"\n\t\t Fecha de Ingreso    : "<<empleado.fingreso;
        cout<<"\n\t\t Fecha de Retiro     : "<<empleado.freti;
        cout<<"\n\t\t/-----/\n";
        getch();
    }
}

/*-----FUNCION CONSULTA DE H.VIDA X APELLIDOS & NOMBRE-----*/
void BusqXapellido()
{
    char ape[30],ape2[30],nom[50];
    int numReg;
    fstream Archivo;
    Archivo.open(ArchivoHV,ios::in|ios::out|ios::binary);
    if (!Archivo){
        Marco3();
    }
    else{
        numReg=0;
        clrscr();
        Marco1();
        cout<<"\n\n\t\t ----- CONSULTA DE HOJA DE VIDA -----";
        cout<<"\n\n\t\tDIGITE EL 1er. APELLIDO: ";
        gets(ape);
        cout<<"\n\n\t\tDIGITE EL 2do. APELLIDO: ";
        gets(ape2);
        cout<<"\n\n\t\tDIGITE EL 1er. NOMBRE      : ";
        gets(nom);
        clrscr();
        do{
            numReg++;
            Archivo.read( (char *)&empleado, sizeof(empleado) );
            if(Archivo.eof()){
                Marco6();
                break;
            }
        }
        if((strcmp(ape,empleado.apellido1)==0)&&(strcmp(ape2,empleado.apellido2)==0)&&(strcmp(
nom,empleado.nombre1)==0)){
            Mostrar();
            cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
            break;
        }
    }while(!Archivo.eof());
    Archivo.close();
}

/*-----FUNCION CONSULTA DE H.VIDA X N DE CEDULA-----*/
void BusqXcedula()
{
    char ced[10];
    int numReg=0;
    clrscr();
    fstream Archivo;
    Archivo.open(ArchivoHV,ios::in|ios::out|ios::binary);
    if (!Archivo){
        Marco3();
    }
    else{
        Marco1();
        cout<<"\n\n\t\t ----- CONSULTA DE HOJA DE VIDA -----";
        cout<<"\n\n\t\tDIGITE EL N DE CEDULA: ";
        gets(ced);
        clrscr();
        do{
            numReg++;
            Archivo.read( (char *)&empleado, sizeof(empleado) );
            if(Archivo.eof()){
                Marco6();
                break;
            }
        }
        if(strcmp(ced,empleado.cedula)==0){
            Mostrar();
            cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
            break;
        }
    }
}

```

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

    }
    }while(!Archivo.eof());
    Archivo.close();
}
}

/*-----FUNCION CONSULTA DE H.VIDA X N DE CODIGO-----*/
void BusqXCodigo()
{
    int CodigoE;
    int numReg;
    fstream Archivo;
    Archivo.open(ArchivoHV, ios::in|ios::out|ios::binary);
    if (!Archivo){
        Marco3();
    }
    else{
        numReg=0;
        clrscr();
        Marco1();
        cout<<"\n\n\t\t ----- CONSULTA DE HOJA DE VIDA -----";
        cout<<"\n\n\t\tDIGITE EL CODIGO DEL EMPLEADO: ";
        cin>>CodigoE;
        clrscr();
        do{
            numReg++;
            Archivo.read( (char *)&empleado, sizeof(empleado) );
            if(Archivo.eof()){
                Marco6();
                break;
            }
            if(CodigoE==empleado.codigoHV){
                Mostrar();
                cout<<resetiosflags(ios::left)<<setiosflags(ios::fixed | ios::showpoint);
                break;
            }
        }while(!Archivo.eof());
        Archivo.close();
    }
}

/*-----FUNCION MODIFICA LOS DATOS DE LAS HOJAS DE VIDA-----*/
void Modificar()
{
    int numReg=0,CodigoE;
    char res1,res2,res3;
    clrscr();
    fstream Archivo;
    Archivo.open(ArchivoHV, ios::in|ios::out|ios::binary);
    if (!Archivo){
        Marco3();
    }
    else{
        Marco1();
        cout<<"\n\n\t\t DIGITE EL CODIGO DEL EMPLEADO [0]-SALIR: ";
        textattr(10);cprintf("[ ]\b\b");
        cin>>CodigoE;
        textattr(7);
        clrscr();
        do{
            numReg++;
            Archivo.read( (char *)&empleado, sizeof(empleado) );
            if(Archivo.eof()){
                Marco6();
                break;
            }
            if(CodigoE==empleado.codigoHV){
                clrscr();
                Marco1();
                cout<<"\n\n\t\t MODIFICAR DATOS PERSONALES? Si [S] / No [N]:";
                textattr(10);cprintf("[ ]\b\b");
                res1=getch();
                textattr(7);
                if(res1=='s' || res1=='S'){
                    cout<<"\n\n\t\t MODIFICAR HOJA DE VIDA ";
                    cout<<"\n\n\t\t-----DATOS PERSONALES-----\n";
                    cout<<"\n\n\t\tler. Apellido: ";
                    gets(empleado.apellido1);
                    cout<<"\n\n\t\t2do. Apellido: ";
                    gets(empleado.apellido2);
                }
            }
        }while(1);
    }
}

```





Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

[illegible]

Software de Almacenamiento de Hojas de Vida	Versión: 2.0
Código Fuente	Fecha: 25/10/2005

```

/*-----FUNCION RESETEA EL ARCHIVO HVIDA.DAT-----*/
void Reset()
{
    int opc;
    Marco2();
    cout<<"\n\t\t";
    cout<<"\n\t\t ALERTA:!ELIMINARA TODAS LAS DE HOJAS DE VIDA!";
    cout<<"\n\t\t";
    cout<<"\n\t\t\u0096";
    cout<<"\n\t\t          ESTA SEGURO? SI [S] / No [N] :";
    textattr(10);cprintf("[ ]\b\b");
    opc=getch();
    textattr(7);
    if(opc=='s' || opc=='S'){
        ofstream arch;
        arch.open ("HVIDAS.DAT", ios::out);
        arch.close();
        Marco2();
        cout<<"\n\t\t";
        cout<<"\n\t\t          ARCHIVO RESETEADO!!!!";
        cout<<"\n\t\t";
        cout<<"\n\t\t\u0096";
        PAUSA;
    }
}

/*-----FUNCION REGISTRA USUARIO - LOGIN & PASSWORD-----*/
void RegistrarU()
{
    char res;
    clrscr();
    ofstream Archivo;
    Archivo.open(ArchivoUSU, ios::ate);
    if (!Archivo){
        Marco3();
    }
    else{
        Marcol();
        cout<<"\n\t\t REGISTRAR NUEVO USUARIO ";
        cout<<"\n\t\t 1er. Apellido: ";
        gets(usuario.apellido1U);
        cout<<"\t\t 2do. Apellido: ";
        gets(usuario.apellido2U);
        cout<<"\t\t 1er. Nombre: ";
        gets(usuario.nombre1U);
        cout<<"\t\t 2do. Nombre: ";
        gets(usuario.nombre2U);
        cout<<"\t\t Digite un Login: ";
        gets(usuario.login);
        do{
            cout<<"\t\t Digite un Password (6-10 CARACTERES):";
            gets(usuario.password);
        }while((strlen(usuario.password)>=11)|| (strlen(usuario.password)<=5));
        //strlen password entre 6 a 11 caracteres alfanumericos
        cout<<"\n\t\tDesea Guardar Usuario? [S]-GUARDAR";
        textattr(10);cprintf("[ ]\b\b");
        res=getch();
        textattr(7);
        if(res=='s' || res=='S'){
            Archivo.write((char *)&usuario, sizeof(usuario));
            Marco4();
        }
        else{
            Marco5();
        }
    }
    Archivo.close();
}

/*----- FUNCIONES VENTANAS UTILIZADAS -----*/
void Marcol()
{
    cout<<"\n\t\t\u0096";
    cout<<"\n\t\t";
    cout<<"\n\t\t          HVIDA V.02";
    cout<<"\n\t\t";
    cout<<"\n\t\t\u0096";
}

/*-----*/

```

