

**MANUAL**

**DE**

**PROGRAMADOR**

## 1.Requerimientos :

Programa de programación de lenguaje tipo c se en su ultima version y mas estable :  
DEV-C++ 5.11

Puede ser utilizado en cualquier de estos sistemas operativos :  
Windows XP / Vista / Windows 7 / Windows 8 / Windows 10.  
Linux.  
Macos.

## 2. Librerias

```
#include <time.h>
```

Utilizamos esta librería para manipular y formatear la fecha y hora del sistema.

```
#include <iostream>
```

Esta librería la utilizamos para operaciones de entrada/salida

```
#include <fstream>
```

Esta librería se utilizó para el manejo de ficheros.

```
#include <windows.h>
```

Esta libreria es importante por que que contiene las declaraciones de todas las funciones de la biblioteca

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

Esta librería contiene los prototipos de funciones de C para gestión de memoria dinámica, control de procesos y otras.

```
#include <conio.h>
```

Esta librería fue utilizada mayor mente para la utilización de cin y cout

```
#include <string>
```

Esta librería fue mayor mente utilizada para el manejo de cadenas



```
#include <cstdlib>
```




















Esta librería se utilizó para la gestión de memoria.

```
#include <vector>
```

**Esta librería se utilizo para el manejo de vectores “arrays”**

### 3.Estructura Raiz.

  **User** : struct

-  **agregar** (ofstream &Es) : void
-  **baja** (ifstream &Lec) : void
-  **buscarcli** (ifstream &Lec) : void
-  **Encabezado** () : void
-  **factura** () : void
-  **fechamod** (ofstream &mo) : void
-  **Fecha\_baja** (ofstream &ba) : void
-  **gotoxy** (int x, int y) : void
-  **Login** () : void
-  **Menu** () : void
-  **modificarRe** (ifstream &Lec) : void
-  **NewUser** (bool su) : void
-  **OcultarCursor** () : void
-  **Verbajas** (ifstream &ve) : void
-  **Verdatos** (ifstream &Lec) : void
-  **Verif** (char nombre[], char clave[])
-  **vizumod** (ifstream &vi) : void
-  **Activo** : User
-  **Datos** : User

## 4.Funciones utilizadas.

### 4.1Funcion gotoxy:

La utilizamos para manipular la posición en la pantalla de lo que nuestra aplicación envíe de salida, con esta función logramos crear una interfaz bastante llamativa, ordenada e intuitiva.

```
void gotoxy(int x,int y){ //Manejo de coordenadas
    HANDLE hcon;
    hcon=GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X=x; dwPos.Y=y;
    SetConsoleCursorPosition(hcon,dwPos);
}
```

```
#define ARR GetAsyncKeyState(VK_UP)
#define ABJ GetAsyncKeyState(VK_DOWN)
#define ENT 13 //enter
#define RET 8 //retroceso
```

También es importante mencionar esta estructura la cual nos deja manipular la consola con las flechas y la tecla enter de nuestro teclado

## 4.2 función IF-THEN-ELSE y función while do

tiene que proporcionar un elemento o condición que sirva para probar y valorar si la expresión se cumple o no.

Los ciclos do-while son una estructura de control cíclica, los cuales nos permiten ejecutar una o varias líneas de código de forma repetitiva sin necesidad de tener un valor inicial e incluso a veces sin siquiera conocer cuando se va a dar el valor final

```
while(caracter=getch()){
    if(caracter==END){
        CTmp[cont]='\0';
        break;
    }
    else{
        if(caracter!=RET){
            CTmp[cont]=caracter;
            cout<<"*";
            cont++;
        }
        else{
            if(cont>0){
                cout<<"\b \b";
                cont--;
            }
        }
    }
}

gotoxy(25,11); cout<<"Confirme contrase"<<char(164)<<"a: ";

fflush(stdin);
cont=0;
```

utilizacion de while

utilizacion if-then esle

Se aplica las funciones mencionadas ya que valora si la función se cumple o no . Y tambien aplica por que la función while se declaro para que ejecute las lineas del codigo de “carácter”

```

    if(Verif(nom, contra)){
        ingresa=true;
    } else{
        intento++;
        if(intento<3){
            printf("\033[0;40;31m"); //secuencia ansi que cambia el color del texto a rojo
            gotoxy(27, 12); cout<<"Usuario y/o clave incorrectos";
            printf("\033[0m"); //secuencia ansi que regresa el color del texto al original
            Sleep(1000);
        } else{
            printf("\033[0;40;31m");
            gotoxy(27, 12); cout<<"Numero de intentos excedido!!";
            printf("\033[0m");
            Sleep(1500);
            exit(1);
        }
        ingresa=false;
    }
}while(!ingresa && intento<3);

```

ya que si se ingresa la contraseña incorrecta o si se excede el número de intentos nos impedirá que la clave o usuario es contraseña y no nos dejará ingresar al menú.

```

if(ingresa){
    strcpy(Activo.nombre, Datos.nombre);
    strcpy(Activo.clave, Datos.clave);
    Activo.Super=Datos.Super;
    Activo.Altas=Datos.Altas;
    Activo.Bajas=Datos.Bajas;
    Activo.Cambios=Datos.Cambios;
    Activo.Factura=Datos.Factura;

    system("cls");
    gotoxy(30,10); cout<<"ACCEDIENDO AL SISTEMA";

    gotoxy(20,12);
    printf("\033[0;40;32m");
    for(int i=0; i<41; i++){
        cout<<char(219);
        Sleep(50);
    }
    printf("\033[0m");

    Menu();
}

```

Pero en el caso de que ingresemos los datos correctos nos imprime que está accediendo al sistema y nos dejará ingresar al menú.

### 4.3 Funcion Strcpy

Esta función nos sirvió para copiar el contenido de una cadena a otra

```
strcpy(Activo.nombre, Datos.nombre);  
strcpy(Activo.clave, Datos.clave);
```

Lo que hay en datos.nombre se copia en activo.nombre

### 3.4 Funcion Switch

que se utiliza para seleccionar una de entre múltiples alternativas.

```
cout<<"      BUSQUEDA DE REGISTROS CLIENTES      "<<endl<<endl;  
cout<<"1. BUSCAR POR NOMBRE"<<endl;  
cout<<"2. BUSCAR POR DPI"<<endl;  
cout<<"3. BUSCAR POR TELEFONO"<<endl;  
cout<<"4. BUSCAR POR PRODUCTO"<<endl;  
cout<<"INGRESE LA OPCION: ";cin>>opcion;
```

```
switch (opcion){  
    case 1: // Busqueda por nombre  
        cout<<"INGRESE EL NOMBRE: ";cin>>nomaux;  
        cout<<"      "<<endl;  
  
        system("cls");  
        gotoxy(30,10); cout<<"BUSCANDO LOS DATOS";  
        gotoxy(20,12);  
        printf("\033[0;40;32m");  
  
        for(int i=0; i<41; i++){ //largo linea carga  
            cout<<char(219);  
            Sleep(50);  
        }
```

utilizacion de la  
funcion switch

Este es un ejemplo de la función switch la cual utilizamos para crear uno de los menús de nuestro programa en este caso el de búsqueda

```
case 2: //Busqueda por dpi  
    cout<<"INGRESE EL DPI: ";cin>>dpiaux;  
    cout<<"      "<<endl;  
  
    system("cls");  
    gotoxy(30,10); cout<<"BUSCANDO LOS DATOS";  
    gotoxy(20,12);  
    printf("\033[0;40;32m");  
  
    for(int i=0; i<41; i++){ //Largo linea carga  
        cout<<char(219);  
        Sleep(50);  
    }  
  
    printf("\033[0m");  
    Encabezado();  
  
    Lec>>nom;  
    while(!Lec.eof() && !encontrado){  
        Lec>>dire;  
        Lec>>muni;  
        Lec>>depa;  
        Lec>>telef;
```

utilizacion de funcion  
switch

Como podemos visualizar esto nos ayuda a seleccionar entre varias opciones.



#### 4.4 sentencia Struct

se utiliza para seleccionar una de entre múltiples alternativas.

```
struct User{  
    char nombre[25], clave[8];  
    int Super, Altas, Cambios, Bajas, Factura;  
};  
struct User Datos;           //estructura para la creaci  
struct User Activo;         //estructura para traslaci
```

En nuestro programa utilizamos esta sentencia para crear nuevos usuarios y para tener almacenados los datos para luego guardarlos en nuestro documento txt.

## 4.5 funciones con punteros tipo file

sirve para abrir y crear ficheros en disco.

utilizamos esta función para abrir, leer, escribir y cerrar archivos txt por ejemplo :

```
LE*comp;  
mp=fopen("Usuarios.txt", "r");
```

archivo utilizando fopen

en este ejemplo estamos abriendo el

```
} fread(&Datos, sizeof(Datos), 1, usuarios);  
}
```

se lee el archivo

```
fclose(usuarios);
```

se cierra el archivo

Luego de  
que el  
archivo se  
abre se lee y  
luego se  
cierra

```
fflush(stdin);
```

También se declara aca comp fflush sirve para  
forzar la salida de los datos acumulados en el  
buffer de salida del fichero para mejorar el manejo  
de ficheros

```
ba.open("bajas.txt", ios::out | ios::app);  
  
cout<<"Ingrese el nombre (Para tener referencia)"<<endl;  
cin>>Nom;  
cout<<"Ingrese el producto que vendio"<<endl;  
cin>>Pro_v;  
cout<<"Ingrese la fecha de baja"<<endl;  
cin>>fecha_baja;  
ba<<Nom<<" "<<Pro_v<<" "<<fecha_baja <<endl;  
ba.close();
```

También podemos visualizar el  
uso de esta función donde abrir  
para ingresar datos y luego  
cerrar el archivo.

## 4.6.

También es importante mencionar que se combinaron varias funciones para crear estructuras como la que se muestra, se utilizó la función gotoxy y la función for para poder crear una pantalla de carga .

```
system("cls");
gotoxy(30,10); cout<<"ACCEDIENDO AL SISTEMA";

gotoxy(20,12);
printf("\033[0;40;32m");
for(int i=0; i<41; i++){
    cout<<char(219);
    Sleep(50);
}
printf("\033[0m");

Menu();
```

```

for(int i=0; i<41; i++){ //largo linea carga
    cout<<char(219);
    Sleep(50);
}

printf("\033[0m");
Encabezado();
gotoxy(10,9);cout<< "
gotoxy(10,10);cout<<"Fecha de emision:"<<" "<<f_fac<<endl;
gotoxy(39,10);cout<<"Nombre Cliente:"<<"nom_c<<endl;
gotoxy(65,10);cout<<"Producto:"<<"pro<<endl;
gotoxy(10,11);cout<<"Precio:"<<" "<<"Q"<<pre<<endl;
gotoxy(10,12);cout<<"Cantidad:"<<" "<<can<<endl;
gotoxy(10,13);cout<<"Monto total de producto"<<" "<<"Q"<<tol<<endl;
gotoxy(10,14);cout<<"IVA incluido"<<" "<<"Q"<<iva<<endl;
gotoxy(10,15);cout<<"
gotoxy(42,15);cout<<"Monto total de la factura"<<" "<<"Q"<<total<<endl;
gotoxy(10,16);cout<<"
gotoxy(86,10);cout<<"<<endl;
gotoxy(86,11);cout<<"<<endl;
gotoxy(86,12);cout<<"<<endl;
gotoxy(86,13);cout<<"<<endl;
gotoxy(86,14);cout<<"<<endl;
gotoxy(86,10);cout<<"<<endl;
cout<<" "<<endl;

```

En esta imagen también se puede visualizar en esta imagen la cual es la creación de una factura en la cual observamos las funciones for y gotoxy

