



**UNIVERSIDAD AUTONOMA  
DE AGUASCALIENTES**

**CENTRO DE CIENCIAS BASICAS**

**DEPARTAMENTO DE SISTEMAS  
ELECTRÓNICOS**

Ingeniería en sistemas computacionales

**Programación I**

ESTELA LIZBETH MUÑOZ ANDRADE

**PROYECTO "FRECUENCIA DE VENTAS"**

**2ºA**

**Manual de Usuario Proyecto ventas**

Ricardo Salvador Heredia Ballin  
212541

Jesús Maximiliano González Frías  
263834

**Aguascalientes, Ags.**

**5 de julio de 2020**

## **I. Introducción**

Este programa está hecho para ayudar a tener un pequeño control dentro de una empresa para registrar los datos de los tipos de empleados que laboran ahí y, a su vez, registrar la cantidad de dinero que estos han recaudado a lo largo de un año. Estos datos se registran en un archivo de texto que el programa genera automáticamente, permitiendo acceder a la información por medio del mismo programa. Finalmente, el programa genera una gráfica a partir de la cantidad de ventas registradas por el usuario y la muestra en pantalla.

## **II. Descripción del problema**

El problema que buscamos resolver con nuestro programa es uno que en algunas empresas es muy común, el cual es la falta de organización a la hora de querer consultar cierta información de los trabajadores, así como de su desempeño en cada mes del año. Creemos que nuestro programa resuelve ese problema de forma gratificante ya que organiza de forma breve y concisa la información registrada.

## **III. Objetivo del proyecto**

El objetivo de este proyecto es entregar un software capaz de almacenar información acerca de un cierto número de empleados y, basándose en esa información, generar un reporte y una gráfica que muestre la cantidad de ventas que cada empleado ha realizado en un plazo de un año.

#### IV. Requerimientos

Los requisitos mínimos que debe tener una computadora para ejecutar exitosamente nuestro programa elaborado en el lenguaje C/C++ son:

- Microsoft Windows 95, 98, NT 4, 2000, XP
- 8 MB de RAM con un archivo de intercambio grande
- Procesador compatible Intel a 100 Mhz
- 30 MB de espacio libre en el disco duro

Sin embargo, para la generación de la gráfica es necesaria una breve instalación de una librería denominada “graphics.h” siendo esta una parte fundamental para tener en cuenta si es que se desea aprovechar al máximo las posibilidades que logra nuestro programa.

Los requerimientos mínimos para ejecutar Dev-C++ son:

- Microsoft Windows 95, 98, NT 4, 2000, XP, o superior
- 32 MB RAM
- 100 Mhz Intel compatible CPU
- 30 MB de espacio libre de disco

Los requerimientos aconsejados para Dev-C++ son:

- Microsoft Windows 2000, XP, o superior
- 64 MB RAM
- 400 Mhz Intel compatible CPU
- 200 MB de espacio libre de disco

## V. Ingreso al sistema

A continuación, se mostrarán los pasos a seguir para ejecutar nuestro programa:

### 1. Menú General

Consta de 4 entradas enumeradas 1.- Empleados, en las cuales se imprime en pantalla la lista de empleados con su código, la fecha de registro y sus ventas anuales. 2.- Ingresar datos, en este apartado es donde se pueden introducir los datos de los empleados dependiendo del tipo de empleado y donde se pueden hacer las consultas. 3.- Mostrar gráficos es donde se muestran las gráficas de las ventas por empleado, donde se muestran del mayor monto por mes al menor. 4.- Salir sirve para cerrar la ventana. Para poder ingresar a cualquiera de estas funciones es necesario teclear su número correspondiente y presionar la tecla enter.

```
Menu Empresarial.  
1.- Empleados.  
2.- Ingresar Datos.  
3.- Mostrar Gráficos.  
4.- Salir.  
Ingrese una opción:
```

## 2. Empleados

Después de acceder a sección solo se imprimirán los datos de la siguiente forma.

### DATOS DE LOS EMPLEADOS

Código del empleado: 212541  
Nombre del empleado: ricks  
FECHA  
6 / 8 / 2020

El monto de la venta del mes 1 es: 124  
El monto de la venta del mes 2 es: 321  
El monto de la venta del mes 3 es: 289  
El monto de la venta del mes 4 es: 99  
El monto de la venta del mes 5 es: 55  
El monto de la venta del mes 6 es: 32  
El monto de la venta del mes 7 es: 205  
El monto de la venta del mes 8 es: 123  
El monto de la venta del mes 9 es: 541  
El monto de la venta del mes 10 es: 98  
El monto de la venta del mes 11 es: 67  
El monto de la venta del mes 12 es: 10

## 3. Ingresar Datos

Al ingresar a este menú se mostrarán 5 opciones, 3 de ellas son similares ya que sólo cambia el tipo de empleado que se registra, la 3.- CONSULTA es donde se realizaran las búsquedas con datos en específico, al acceder a ella se abrirá otro sub-menú del cual se hablará en posteriormente. Y el 4.- REGRESAR sirve para regresar al menú anterior.

### TIPOS DE EMPLEADOS

Administrador....0  
Gerente....1  
Vendedor....2  
CONSULTA....3  
Regresar....4  
Ingrese una opción:

#### a. CONSULTA

Aquí es donde se podrán hacer búsquedas dentro de los datos almacenados son de 3 tipos por código de empleado, por fecha y monto de la venta, con el código de empleado se podrá obtener la información de la siguiente dentro del archivo .txt generado junto a la carpeta donde está el archivo ejecutable.

```
Código encontrado: 212541
Nombre: ricks
FECHA
Día: 6          Mes: 8          Año: 2020

Monto de venta del mes 1: 124,00
Monto de venta del mes 2: 321,00
Monto de venta del mes 3: 289,00
Monto de venta del mes 4: 99,00
Monto de venta del mes 5: 55,00
Monto de venta del mes 6: 32,00
Monto de venta del mes 7: 205,00
Monto de venta del mes 8: 123,00
Monto de venta del mes 9: 541,00
Monto de venta del mes 10: 98,00
Monto de venta del mes 11: 67,00
Monto de venta del mes 12: 10,00
```

Para buscar por fecha se introduce día, mes y año y te dará el reporte de la misma manera en el archivo, pero en este caso mostrará lo encontrado en base a cada dato introducido.

```
Día encontrado: 6
Código: 212541
Nombre: ricks
FECHA
                Mes: 8                Año: 2020

Monto de venta del mes 1: 124,00
Monto de venta del mes 2: 321,00
Monto de venta del mes 3: 289,00
Monto de venta del mes 4: 99,00
Monto de venta del mes 5: 55,00
Monto de venta del mes 6: 32,00
Monto de venta del mes 7: 205,00
Monto de venta del mes 8: 123,00
Monto de venta del mes 9: 541,00
Monto de venta del mes 10: 98,00
Monto de venta del mes 11: 67,00
Monto de venta del mes 12: 10,00
```

La última es la búsqueda por monto de venta donde se encontrarán todos los montos que se encuentren en base al introducido por el usuario.

Monto de venta del mes 6: 123,00

Código: 1234

Nombre: Max

FECHA

Día: 6

Mes: 6

Año: 2001

Monto de venta del mes 2: 123,00

Código: 546

Nombre: Quinn

FECHA

Día: 5

Mes: 8

Año: 2019

Monto de venta del mes 3: 123,00

Código: 546

Nombre: Quinn

FECHA

Día: 5

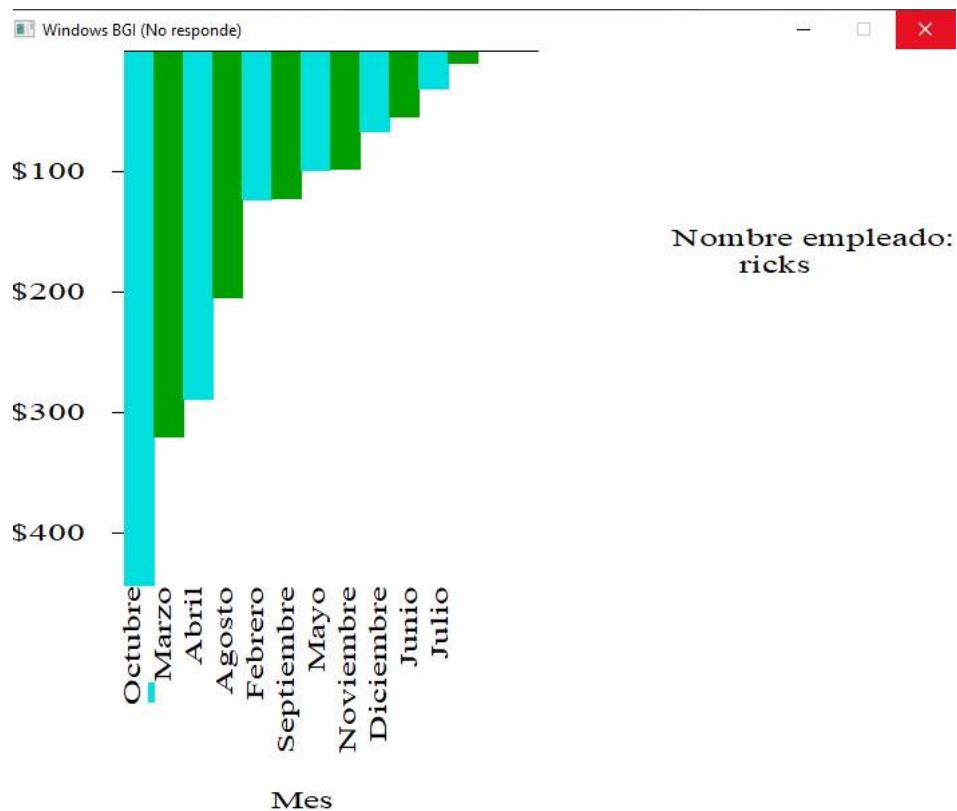
Mes: 8

Año: 2019



## b. GRÁFICAS

En este apartado se mostrarán graficas de todos los empleados registrados junto con los datos de sus ventas por año ordenándolos de mayor a menor de la siguiente manera:



Para acceder a las demás gráficas, solo hay que presionar la tecla enter en la consola para que continúe mostrando las que estén registradas en el archivo.

## VI. Bibliografía

Las principales fuentes de ayuda para la elaboración de este proyecto fueron los videos:

[https://www.youtube.com/watch?v=xMGP8nXA\\_GY&t=269s](https://www.youtube.com/watch?v=xMGP8nXA_GY&t=269s)

**Título del video:** 3. Funciones de Graphics.h | Guía de graphics.h | C++

**Autor del video:** Jibaru

**Fecha de publicación:** 8 de junio de 2019

[https://www.youtube.com/watch?v=P\\_mqwdzMcFo](https://www.youtube.com/watch?v=P_mqwdzMcFo)

**Título del video:** 4.2. Graficar un plano cartesiano - Parte 1 | Guía de graphics.h | C++

**Autor del video:** Jibaru

**Fecha de publicación:** 8 de junio de 2019

Así como las siguientes páginas web:

*Biblioteca graphics Borland® C.* (s. f.). c.conclase.net. Recuperado 20 de junio de 2020, de <http://c.conclase.net/borland/index.php?borlandlib=graphics#inicio>

Macias, A. (s. f.). *Requerimientos Para Instalar C++ - (DevC++) y Reglas.* tronand0genios.blogspot.com. Recuperado 4 de julio de 2020, de <https://tronand0genios.blogspot.com/p/requerimientos-para-instalar-de.html>

*Gráficos Usando graphics.h.* (2016, 7 marzo). Recuperado 12 de junio de 2020, de

[http://chuwiki.chuidiang.org/index.php?title=Gr%C3%A1ficos\\_Usando\\_graphics.h](http://chuwiki.chuidiang.org/index.php?title=Gr%C3%A1ficos_Usando_graphics.h)

## VII. Anexos

A continuación, se anexa todo el código del proyecto:

```
// AUTORES DEL CÓDIGO:
//---JESUS MAXIMILIANO GONZALEZ FRIAS
//---RICARDO SALVADOR HEREDIA BALLIN
//FECHA DE ENTREGA: 5 DE JULIO DE 2020

// DESCRIPCIÓN GENERAL DEL CÓDIGO
// ESTE CÓDIGO REALIZA UNA CAPTURA DE LOS EMPLEADOS DE UNA
// EMPRESA SOLICITANDO UNOS DATOS DETERMINADOS, ESTO CON EL
// FIN DE TENER ALMACENADA LA INFORMACIÓN DE ESTAS PERSONAS
// EN UN ARCHIVO DE TEXTO.
// A SU VEZ, ESTE CÓDIGO TIENE LA OPCIÓN DE GENERAR UNA
// GRÁFICA UTILIZANDO LA INFORMACIÓN REGISTRADA EN EL ARCHIVO
// YA MENCIONADO ANTERIORMENTE.
// LA REALIZACIÓN DE ESTA GRÁFICA FUE POSIBLE GRACIAS A LA
// LIBRERÍA DE C++ LLAMADA "GRAPHICS.H".

// LIBRERIAS UTILIZADAS
#include <iostream> // LIBRERÍA BASE DE C++
#include <stdio.h> // LIBRERÍA DE ENTRADA Y SALIDA ESTÁNDAR
#include <string.h> // LIBRERÍA PARA EL USO DE CARACTERES
#include <graphics.h> // LIBRERÍA DE GRÁFICOS
#include <locale.h> // LIBRERÍA PARA PONER ACENTOS EN EL
// COMPILADOR

//INSTRUCCIÓN POR DEFECTO DE C++
using namespace std;

//DECLARACIÓN DE UNA CONSTANTE
const int TAM = 12;

//FUNCIÓN POR DEFECTO DE LA LIBRERÍA GRAPHICS.H
void gotoxy(int x,int y){
    HANDLE hcon;
    hcon = GetStdHandle(STD_OUTPUT_HANDLE);
    COORD dwPos;
    dwPos.X = x;
    dwPos.Y= y;
    SetConsoleCursorPosition(hcon,dwPos);
}

//ENUMERACIÓN UTILIZADA
```



```

cout << "\n\t\t\t\t\t3.- Mostrar Gráficos.";
cout << "\n\t\t\t\t\t4.- Salir.";
cout << "\n\t\t\t\t\tIngrese una opción: ";

```

//CÓDIGO UTILIZADO PARA VALIDAR SI SE INGRESA UN CARACTER

```

if(scanf("%d",&op) != 1){
    cout << "\n\t\t\t\t\tERROR!!! EL VALOR DEBE SER UN
NUMERO\n";

```

```

    while(getchar() != '\n');
    system("pause>null");
    system("cls");

```

```

} else if(op == 1 || op == 2 || op == 3 || op == 4){

```

```

switch(op){

```

```

    case 1:

```

```

        //INSTRUCCIÓN PARA LIMPIAR LA PANTALLA

```

```

        system("cls");

```

```

        //LLAMADA DE LA FUNCIÓN

```

```

        mostrarEmpleados(arch);

```

```

        cout << endl;

```

AL SER EJECUTADO

```

        system("pause");

```

```

        system("cls");

```

```

        break;

```

```

    case 2:

```

```

        system("cls");

```

```

        //LLAMADA DE LA FUNCIÓN

```

```

        capturaEmpleado(op, arch, arch2);

```

```

        cout << endl;

```

```

        system("pause");

```

```

        system("cls");

```

```

        break;

```

```

    case 3:

```

```

        //LLAMADA DE LA FUNCIÓN

```

```

        graficas(arch);

```

```

        system("cls");

```

```

        break;

```

```

    }

```

```

} else{

```



```

    cout << "\nDATOS DE LOS EMPLEADOS";
    //INSTRUCCIÓN PARA COLOCARSE AL INICIO DEL ARCHIVO
    rewind(arch);
    //INSTRUCCIÓN PARA EJECUTAR CÓDIGO MIENTRAS NO SEA FIN DE
    ARCHIVO
    while(!(feof(arch))){
        //EXTRAER DATOS DEL ARCHIVO
        fscanf(arch,"%d", &mostrar.codigoEmp);
        fscanf(arch,"%s", mostrar.nomEmpleado);
        fscanf(arch,"%d", &mostrar.FechaVta.dd);
        fscanf(arch,"%d", &mostrar.FechaVta.mm);
        fscanf(arch,"%d", &mostrar.FechaVta.aa);

        //EXTRAER LOS DATOS REGISTRADOS EN EL VECTOR
        DINÁMICO
        for(int i=0; i<TAM; i++){
            fscanf(arch,"%f", &monto[i]);
        }

        //MOSTRAR LOS DATOS EN EL COMPILADOR DEL PROGRAMA
        cout << endl;
        cout << "\nCódigo del empleado: " << mostrar.codigoEmp;
        cout << "\nNombre del empleado: " << mostrar.nomEmpleado;
        cout << "\nFECHA\n";
        cout << mostrar.FechaVta.dd << " / " << mostrar.FechaVta.mm << " /
" << mostrar.FechaVta.aa << endl;

        //LLAMADA DE LA FUNCIÓN
        mostrarMon(monto);
        cout << endl;
        system("pause>null");
    }
    //CIERRE DEL ARCHIVO DE TEXTO
    fclose(arch);
}

//FUNCIÓN PARA MOSTRAR EL CONTENIDO DEL VECTOR DINÁMICO EN
EL COMPILADOR
void mostrarMon(float *monto){
    for(int i=0; i<TAM; i++){
        cout << "\nEl monto de venta del mes " << i+1 << " es: " <<
*(monto+i);
    }
}

```

//FUNCIÓN PARA CAPTURAR TODOS LOS TIPOS DE EMPLEADOS QUE PERMITE EL PROGRAMA

```
void capturaEmpleado(int op, FILE *arch, FILE *arch2){
    op = 0;
    Empleado registro;

    float *monto = new float[TAM];
    monto = limpiarMonto(monto);

    do{
        cout << "\n\t\t\t\t\t***TIPOS DE EMPLEADOS***\n\n";
        //CÓDIGO PARA HACER USO DE LA ENUMERACIÓN
        DECLARADA ANTERIORMENTE
        for(int i=0; i<5; i++){
            cout << "\t\t\t\t\t" << menuEmp[i] << "...." << i << endl;
        }

        cout << "\t\t\t\t\tIngrese una opción: ";
        if(scanf("%d",&op) != 1){
            cout << "\n\t\t\t\t\tERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            return;
        }

        switch(op){
            case ADMINISTRADOR:
                system("cls");

                arch = fopen("RegistroEmpleados.txt","a+");
                if(arch == NULL){
                    cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
                    return;
                }

                //INSTRUCCIÓN PARA "ACCEDER" AL ELEMENTO DE LA
                ENUMERACIÓN
                cout << "\nHa seleccionado: " << menuEmp[0] << endl;

                cout << "Escribe el código del empleado: ";
                if(scanf("%d",&registro.codigoEmp) != 1){
                    cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
                    while(getchar() != '\n');
                    break;
                }
            }
        }
    }
}
```



```

        cout << "Escribe el nombre del empleado: ";
        //INSTRUCCIÓN PARA LIMPIAR EL BUFFER
        fflush(stdin);
        //INSTRUCCIÓN PARA LEER UNA CADENA DE
CARACTERES INCLUYENDO ESPACIOS
        gets(registro.nomEmpleado);

        cout << "\nFECHA\n";
        cout << "Escribe el día: ";
        if(scanf("%d",&registro.FechaVta.dd) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el mes: ";
        if(scanf("%d",&registro.FechaVta.mm) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el año: ";
        if(scanf("%d",&registro.FechaVta.aa) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        //GRABAR LOS DATOS PROPORCIONADOS EN EL
ARCHIVO
        fprintf(arch, "\n%d %s %d %d %d",registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.dd, registro.FechaVta.mm,
registro.FechaVta.aa);

        //LLAMADA DE LA FUNCIÓN
llenado(arch, monto);

        fclose(arch);

        cout << endl;
        system("pause");
        system("cls");

```

```

        break;

    case GERENTE:
        system("cls");

        arch = fopen("RegistroEmpleados.txt","a+");
        if(arch == NULL){
            cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
            return;
        }

        cout << "\nHa seleccionado: " << menuEmp[1] << endl;

        cout << "Escribe el código del empleado: ";
        if(scanf("%d",&registro.codigoEmp) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el nombre del empleado: ";
        fflush(stdin);
        gets(registro.nomEmpleado);

        cout << "\nFECHA\n";
        cout << "Escribe el día: ";
        if(scanf("%d",&registro.FechaVta.dd) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el mes: ";
        if(scanf("%d",&registro.FechaVta.mm) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el año: ";
        if(scanf("%d",&registro.FechaVta.aa) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');

```

```

        break;
    }

    fprintf(arch, "\n%d %s %d %d %d", registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.dd, registro.FechaVta.mm,
registro.FechaVta.aa);

llenado(arch, monto);

fclose(arch);

cout << endl;
system("pause");
system("cls");
break;

case VENDEDOR:
    system("cls");

    arch = fopen("RegistroEmpleados.txt", "a+");
    if(arch == NULL){
        cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
        return;
    }

    cout << "\nHa seleccionado: " << menuEmp[2] << endl;

    cout << "Escribe el código del empleado: ";
    if(scanf("%d",&registro.codigoEmp) != 1){
        cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
        while(getchar() != '\n');
        break;
    }

    cout << "Escribe el nombre del empleado: ";
    fflush(stdin);
    gets(registro.nomEmpleado);

    cout << "\nFECHA\n";
    cout << "Escribe el día: ";
    if(scanf("%d",&registro.FechaVta.dd) != 1){
        cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
        while(getchar() != '\n');
        break;
    }
}

```

```

        cout << "Escribe el mes: ";
        if(scanf("%d",&registro.FechaVta.mm) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        cout << "Escribe el año: ";
        if(scanf("%d",&registro.FechaVta.aa) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        fprintf(arch, "\n%d %s %d %d %d",registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.dd, registro.FechaVta.mm,
registro.FechaVta.aa);

        llenado(arch, monto);
        fclose(arch);

        cout << endl;
        system("pause");
        system("cls");
        break;

    case CONSULTA:
        system("cls");

        consulta(op,arch,arch2);

        cout << endl;
        system("pause");
        system("cls");
        break;

    case REGRESAR:
        return;

//VALIDAR SI SE INGRESA UNA OPCIÓN QUE NO ESTÉ
DENTRO DEL RANGO DE OPCIONES DEL MENÚ
    default:
        cout << "\nESCRIBA UNA OPCION VALIDA\n";

```

```

        cout << endl;
        system("pause");
        system("cls");
        break;
    }
}while(op != REGRESAR);
}

//FUNCIÓN PARA BUSCAR INFORMACIÓN ESPECÍFICA EN EL ARCHIVO
Y PARA GENERAR UN REPORTE
void consulta(int op, FILE *arch, FILE *arch2){
    op = 0;
    Empleado registro;

    float *monto = new float[TAM];
    monto = limpiarMonto(monto);

    int codigoCons=0, bandera=0, ddBus=0, mmBus=0, aaBus=0, *ptr;
    float montoBus;

    do{
        cout << "\n\t\t\t\t\t***CONSULTA DE DATOS***\n ";
        cout << "\n\t\t\t\t\tConsulta por código de empleado....1";
        cout << "\n\t\t\t\t\tConsulta por fecha.....2";
        cout << "\n\t\t\t\t\tConsulta por monto de venta.....3";
        cout << "\n\t\t\t\t\tREGRESAR.....4";

        cout << "\n\t\t\t\t\tIngrese una opción: ";
        if(scanf("%d",&op) != 1){
            cout << "\n\t\t\t\t\tERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            return;
        }

        switch(op){
            case 1:
                system("cls");
                bandera = 0;
                arch = fopen("RegistroEmpleados.txt","r");
                if(arch == NULL){
                    cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
                    return;
                }

```

//APERTURA DE OTRO ARCHIVO DE TEXTO

```

        arch2 = fopen("ReporteEmpleados.txt","a+");
        if(arch2 == NULL){
            cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
            return;
        }

        ptr = &codigoCons;

        cout << "\nEscribe el código del empleado a buscar: ";
        if(scanf("%d", ptr) != 1){
            cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
            while(getchar() != '\n');
            break;
        }

        rewind(arch);
        while(!feof(arch)){
            fscanf(arch, "%d", &registro.codigoEmp);
            fscanf(arch, "%s", registro.nomEmpleado);
            fscanf(arch, "%d", &registro.FechaVta.dd);
            fscanf(arch, "%d", &registro.FechaVta.mm);
            fscanf(arch, "%d", &registro.FechaVta.aa);

            for(int i=0; i<TAM; i++){
                fscanf(arch,"%f", &monto[i]);
            }

            //CONDICIÓN PARA CONSULTAR EL CÓDIGO
INGRESADO POR EL USUARIO
            if(codigoCons == registro.codigoEmp){
                //INSTRUCCIONES PARA GRABAR LOS DATOS EN
EL ARCHIVO REPORTE
                fprintf(arch2, "\n\nCódigo encontrado: %d",
registro.codigoEmp);
                fprintf(arch2, "\n Nombre del empleado: %s \n FECHA \n
Día: %d \t Mes: %d \t Año: %d \n ",registro.nomEmpleado,
registro.FechaVta.dd, registro.FechaVta.mm, registro.FechaVta.aa);

                for(int i=0; i<TAM; i++){
                    fprintf(arch2, "\nMonto de venta del mes
%d: %.2f", i+1, monto[i]);
                }
                bandera = 1;
            }else{
                bandera = 0;
            }if(bandera == 1){

```

```

        break;
    }
}

//CONDICIÓN PARA MOSTRAR UN MENSAJE AL
USUARIO SI ES QUE SE ENCONTRÓ EL REGISTRO
if(bandera == 1){
    cout << "\nSe encontró el código !!! \n";
    cout << "\n***REPORTE GENERADO***\n";
}else if (bandera == 0){
    cout << "\nNo encontró el código !!! ";
}

//CIERRE DE LOS ARCHIVOS DE TEXTO
fclose(arch);
fclose(arch2);

cout << endl;
system("pause");
system("cls");
break;

case 2:
    system("cls");
    bandera = 0;
    arch = fopen("RegistroEmpleados.txt","r");
    if(arch == NULL){
        cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
        return;
    }

    arch2 = fopen("ReporteEmpleados.txt","a+");
    if(arch2 == NULL){
        cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
        return;
    }

    cout << "\nEscribe el día a buscar: ";
    if(scanf("%d",&ddBus) != 1){
        cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
        while(getchar() != '\n');
        break;
    }

    cout << "\nEscribe el mes a buscar: ";
    if(scanf("%d",&mmBus) != 1){

```

```

        cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
        while(getchar() != '\n');
        break;
    }

    cout << "\nEscribe el año a buscar: ";
    if(scanf("%d",&aaBus) != 1){
        cout << "\nERROR!!! EL VALOR DEBE SER UN
NUMERO\n";
        while(getchar() != '\n');
        break;
    }

    rewind(arch);
    while(!feof(arch)){
        fscanf(arch, "%d", &registro.codigoEmp);
        fscanf(arch, "%s", registro.nomEmpleado);
        fscanf(arch, "%d", &registro.FechaVta.dd);
        fscanf(arch, "%d", &registro.FechaVta.mm);
        fscanf(arch, "%d", &registro.FechaVta.aa);

        for(int i=0; i<TAM; i++){
            fscanf(arch, "%f", &monto[i]);
        }

        //CONDICIÓN PARA CONSULTAR EL DIA, MES Y AÑO
        INGRESADO POR EL USUARIO
        if(ddBus == registro.FechaVta.dd && mmBus ==
registro.FechaVta.mm && aaBus == registro.FechaVta.aa){
            fprintf(arch2, "\n\nDía encontrado: %d
",registro.FechaVta.dd);
            fprintf(arch2, "\n Código: %d \n Nombre del empleado:
%s \n FECHA \n \t\t Mes: %d \t Año: %d \n",registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.mm, registro.FechaVta.aa);

            for(int i=0; i<TAM; i++){
                fprintf(arch2, "\nMonto de venta del mes
%d: %.2f", i+1, monto[i]);
            }

            fprintf(arch2, "\n\nMes encontrado: %d
",registro.FechaVta.mm);
            fprintf(arch2, "\n Código: %d \n Nombre del empleado:
%s \n FECHA \n Día: %d \t\t Año: %d \n",registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.dd, registro.FechaVta.aa);

```



```

        for(int i=0; i<TAM; i++){
            fprintf(arch2, "\nMonto de venta del mes
%d: %.2f", i+1, monto[i]);
        }

        fprintf(arch2, "\n\nAño encontrado: %d
",registro.FechaVta.aa);
        fprintf(arch2, "\n Código: %d \n Nombre del empleado:
%s \n FECHA \n Día: %d \t Mes: %d \n",registro.codigoEmp,
registro.nomEmpleado, registro.FechaVta.dd, registro.FechaVta.mm);

        for(int i=0; i<TAM; i++){
            fprintf(arch2, "\nMonto de venta del mes
%d: %.2f", i+1, monto[i]);
        }

        bandera = 1;
        }else{
            bandera = 0;
        }if(bandera == 1){
            break;
        }
    }

    if(bandera == 1){
        cout << "\nSe encontró la fecha !!! \n";
        cout << "\n***REPORTE GENERADO***\n";
    }else if (bandera == 0){
        cout << "\nNo encontró la fecha !!! ";
    }

    fclose(arch);
    fclose(arch2);

    cout << endl;
    system("pause");
    system("cls");
    break;

case 3:
    system("cls");
    bandera = 0;
    arch = fopen("RegistroEmpleados.txt","r");
    if(arch == NULL){
        cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
        return;
    }
}

```

```

        arch2 = fopen("ReporteEmpleados.txt","a+");
if(arch2 == NULL){
    cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
    return;
}

cout << "\nEscribe el monto a buscar: ";
cin >> montoBus;

rewind(arch);
while(!(feof(arch))){
    fscanf(arch, "%d", &registro.codigoEmp);
    fscanf(arch, "%s", registro.nomEmpleado);
    fscanf(arch, "%d", &registro.FechaVta.dd);
    fscanf(arch, "%d", &registro.FechaVta.mm);
    fscanf(arch, "%d", &registro.FechaVta.aa);

    for(int i=0; i<TAM; i++){
        fscanf(arch,"%f", &monto[i]);
    }

    for(int i=0; i<TAM; i++){
        //CONDICIÓN PARA CONSULTAR LOS MONTOS DE
        VENTA INGRESADOS POR EL USUARIO (REGISTRADOS EN EL VECTOR
        DINÁMICO)
        if(*(monto+i) == montoBus){
            fprintf(arch2, "\nMonto de venta del mes
%d: %.2f", i+1, monto[i]);
            fprintf(arch2, "\n \nCódigo: %d \n Nombre
del empleado: %s \n FECHA \n Día: %d \t Mes: %d \t Año: %d
\n",registro.codigoEmp, registro.nomEmpleado, registro.FechaVta.dd,
registro.FechaVta.mm, registro.FechaVta.aa);
            bandera = 1;
        }
    }

    if(bandera == 1){
        cout << "\nSe encontró el monto de la venta !!! ";
        cout << "\n***REPORTE GENERADO***\n";
        bandera = 0;
    }else if(bandera==0){
        cout << "\nNo se encontró el monto de la venta
!!! ";
    }
}

```

```

        fclose(arch);
        fclose(arch2);

        cout << endl;
        system("pause");
        system("cls");
        break;

    case 4:
        return;

    default:
        cout << "\nESCRIBA UNA OPCION VALIDA";

        cout << endl;
        system("pause");
        system("cls");
        break;
    }
}while(op != 4);
}

```

//FUNCIÓN PARA GENERAR LA GRÁFICA DEL PROGRAMA

```

void graficas(FILE *arch){
    int y1,band=0,flood=100;
    int mesM[12]={1,2,3,4,5,6,7,8,9,10,11,12};
    int auxM;
    int x1=112,x2=90,xM=107;
    char mes[25];
    Empleado reg;
    float *monto = new float[TAM];
    monto = limpiarMonto(monto);
    //INICIALIZAR GRAFICOS
    initwindow(720,600,"Grafica");
    setbkcolor(WHITE);
    cleardevice();

    arch = fopen("RegistroEmpleados.txt","r");
    if(arch == NULL){
        cout << "\nERROR AL ABRIR EL ARCHIVO !!!";
        return;
    }

    rewind(arch);
    // CICLO WHILE SACAR DATOS Y GRAFICAS
    while(!feof(arch)){

```

```

fscanf(arch, "%d", &reg.codigoEmp);
fscanf(arch, "%s", reg.nomEmpleado);
fscanf(arch, "%d", &reg.FechaVta.dd);
fscanf(arch, "%d", &reg.FechaVta.mm);
fscanf(arch, "%d", &reg.FechaVta.aa);
for(int i=0; i<TAM; i++){
    fscanf(arch,"%f", &monto[i]);
}
x1=112;
x2=90;
xM=107;
flood=100;

//IMPPRIMIR LAS BASES DE LA GRAFICA
setcolor(BLACK);
line(90,0,400,0);
line(90,360,90,0);
line(90,270,80,270);//100
line(90,180,80,180);//200
line(90,90,80,90);//300
line(90,360,80,360);//400

//IMPRIMIR EN LA VENTANA
setcolor(BLACK);
settextstyle(6,0,1);
outtextxy(5,350,"$400");
outtextxy(5,260,"$300");
outtextxy(5,170,"$200");
outtextxy(5,80,"$100");

//DATOS
outtextxy(500,130,"Nombre empleado:");
outtextxy(550,150,reg.nomEmpleado);

float aux;
//CICLO FOR PARA DETERMINAR EL MONTO MAYOR EN EL MES
for(int i=1; i<12; i++)
{
    for(int j=0; j<12-i; j++)
    {
        if(monto[j]<monto[j+1])
        {
            aux = monto[j+1];
            monto[j+1] = monto[j];
            monto[j] = aux;

            auxM = mesM[j+1];

```

```

        mesM[j+1] = mesM[j];
        mesM[j] = auxM;
    }
}

//FUNCION PARA GENERAR LAS BARRAS DE LAS GRAFICAS
for(int i=0;i<12;i++){
    y1=regla3(monto,i);
    if(band == 1){
        setcolor(GREEN);
    }
    else{
        setcolor(CYAN);
    }
    rectangle(x1,1,x2,y1);
    if(band == 1){
        setcolor(GREEN);
        setfillstyle(SOLID_FILL, GREEN);
        floodfill(flood,y1/2, GREEN);
        band=0;
    }
    else{
        setcolor(CYAN);
        band=1;
        setfillstyle(SOLID_FILL, CYAN);
        floodfill(flood,y1/2, CYAN);
    }
    x1+=22;
    x2+=22;
    flood+=22;
}
setcolor(BLACK);
settextstyle(6,0,1);
outtextxy(200,550,"Mes");
//FOR PARA IMPRIMIR LOS MESES EN initwindo();
for(int i=0;i<13;i++){
    switch(mesM[i]){
        case 1: strcpy(mes,"Enero");
                break;
        case 2: strcpy(mes,"Febrero");
                break;
        case 3: strcpy(mes,"Marzo");
                break;
        case 4: strcpy(mes,"Abril");
                break;
    }
}

```

```

        case 5: strcpy(mes,"Mayo");
                break;
        case 6: strcpy(mes,"Junio");
                break;
        case 7: strcpy(mes,"Julio");
                break;
        case 8: strcpy(mes,"Agosto");
                break;
        case 9: strcpy(mes,"Septiembre");
                break;
        case 10: strcpy(mes,"Octubre");
                break;
        case 11: strcpy(mes,"Noviembre");
                break;
        case 12: strcpy(mes,"Diciembre");
                break;

    }
    setcolor(BLACK);
    settextstyle(6,1,1);
    outtextxy(xM,400,mes);
    xM+=23;
    }
    for(int i=0;i<12;i++){
        mesM[i]=i+1;
    }
    cout << "ID empleado: " << reg.codigoEmp << endl;

    cout << "Siguiente grafica....." << endl;
    system("pause");
    system("cls");
    cleardevice();
    }
    getch();
    fclose(arch);
}

//FUNCION PARA SACAR LA EQUIVALENCIA DE PUNTOS A PÍXELES
int regla3(float *monto, int i){
    return monto[i]*90/100;
}

```