



SISTEMAS OPERATIVOS 1
PRACTICA1

MAESTRO: JOSUE PEREZ – LUCERO

ALUMNO: IRVYN XICALE CABRERA 201963582

CÓDIGO

REGISTRO.CPP

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

class Registro{
//atributos
public:
    struct Venta{
        char vendedor[25];
        int total;
        int fecha;
        struct Venta *sig;
    }*ventas,*aux;

// metodos
    Registro();
    void registrar(char[],int,int);
    void mostrar();
    void eliminar(int);
    void ordenarVendedor(int);
    void ordenarTotal(int);
    void ordenarFecha(int);

};

Registro::Registro(){
    ventas=NULL;
}

// registrar
void Registro::registrar(char ven[],int tot,int fech){
    Venta *nuevaVenta, *tmp; //se crean nodos auxiliares
    nuevaVenta=(Venta *) malloc(sizeof(Venta)); //se asigna un espacio en memoria para el nodo

//ingresa la informacion del nodo
    strcpy( nuevaVenta->vendedor, ven );
    nuevaVenta->total=tot;
    nuevaVenta->fecha=fech;
    nuevaVenta->sig=NULL;

    if(ventas==NULL){
        ventas=nuevaVenta;
    }else{
        tmp=ventas;
        while(tmp->sig != NULL){
            tmp=tmp->sig;
        }
        tmp->sig=nuevaVenta;
    }
}
```

```

}

tmp->sig=nuevaVenta;
}
}

// mostrar
void Registro::mostrar(){
aux = ventas;
printf("\n\n");
int cont=1;
while(aux != NULL){
printf("%d.- vendedor: %s total:%d fecha:%d\n", cont,aux->vendedor,aux->total,aux->fecha);
aux=aux->sig;
cont++;
}
}

// eliminar
void Registro::eliminar(int eliminar){
Venta *nodoAnterior;
aux = ventas;
int cont=1;
if(eliminar==1){
ventas=ventas->sig;
free(aux);
}else{
while(cont!=eliminar || aux==NULL){
nodoAnterior=aux;
aux=aux->sig;
cont++;
}
if(aux!=NULL){
nodoAnterior->sig=aux->sig;
free(aux);
}
}
}

// ordenar por vendedores
void Registro::ordenarVendedor(int forma){
char tmpV[10];
int tmpT,tmpF;
bool fin=true;
aux=ventas;
if(forma==1){
// ascendente
while (aux->sig!=NULL || fin==false){
if (aux->sig==NULL)

```

```

{
fin=true;
aux=ventas;
}
if(strcmp(aux->vendedor,aux->sig->vendedor)>0){
// guardamos en una variable temporal
strcpy( tmpV, aux->sig->vendedor );
tmpT = aux->sig->total;
tmpF = aux->sig->fecha;

// cambiamos los datos
strcpy( aux->sig->vendedor, aux->vendedor );
aux->sig->total = aux->total;
aux->sig->fecha = aux->fecha;

strcpy( aux->vendedor, tmpV );
aux->total = tmpT;
aux->fecha = tmpF;

fin=false;
}
aux=aux->sig;
}
}
else{
// descendente
while (aux->sig!=NULL || fin==false){
if (aux->sig==NULL)
{
fin=true;
aux=ventas;
}
if(strcmp(aux->vendedor,aux->sig->vendedor)<0){
// guardamos en una variable temporal
strcpy( tmpV, aux->sig->vendedor );
tmpT = aux->sig->total;
tmpF = aux->sig->fecha;

// cambiamos los datos
strcpy( aux->sig->vendedor, aux->vendedor );
aux->sig->total = aux->total;
aux->sig->fecha = aux->fecha;

strcpy( aux->vendedor, tmpV );
aux->total = tmpT;
aux->fecha = tmpF;

fin=false;
}
}
}

```

```
aux=aux->sig;  
}  
}  
}
```

```
// ordenar por total  
void Registro::ordenarTotal(int forma){  
    char tmpV[10];  
    int tmpT,tmpF;  
    bool fin=true;  
    aux=ventas;  
    if(forma==1){  
        // ascendente  
        while (aux->sig!=NULL || fin==false){  
            if (aux->sig==NULL)  
            {  
                fin=true;  
                aux=ventas;  
            }  
            if(aux->total > aux->sig->total){  
                // guardamos en una variable temporal  
                strcpy( tmpV, aux->sig->vendedor );  
                tmpT = aux->sig->total;  
                tmpF = aux->sig->fecha;  
  
                // cambiamos los datos  
                strcpy( aux->sig->vendedor, aux->vendedor );  
                aux->sig->total = aux->total;  
                aux->sig->fecha = aux->fecha;  
  
                strcpy( aux->vendedor, tmpV );  
                aux->total = tmpT;  
                aux->fecha = tmpF;  
  
                fin=false;  
            }  
            aux=aux->sig;  
        }  
    }  
    else{  
        // descendente  
        while (aux->sig!=NULL || fin==false){  
            if (aux->sig==NULL)  
            {  
                fin=true;  
                aux=ventas;  
            }  
            if(aux->total < aux->sig->total){
```

```

// guardamos en una variable temporal
strcpy( tmpV, aux->sig->vendedor );
tmpT = aux->sig->total;
tmpF = aux->sig->fecha;

// cambiamos los datos
strcpy( aux->sig->vendedor, aux->vendedor );
aux->sig->total = aux->total;
aux->sig->fecha = aux->fecha;

strcpy( aux->vendedor, tmpV );
aux->total = tmpT;
aux->fecha = tmpF;

fin=false;
}
aux=aux->sig;
}
}
}

```

```

// ordenar por fecha
void Registro::ordenarFecha(int forma){
char tmpV[10];
int tmpT,tmpF;
bool fin=true;
aux=ventas;
if(forma==1){
// ascendente
while (aux->sig!=NULL || fin==false){
if (aux->sig==NULL)
{
fin=true;
aux=ventas;
}
if(aux->fecha > aux->sig->fecha){
// guardamos en una variable temporal
strcpy( tmpV, aux->sig->vendedor );
tmpT = aux->sig->total;
tmpF = aux->sig->fecha;

// cambiamos los datos
strcpy( aux->sig->vendedor, aux->vendedor );
aux->sig->total = aux->total;
aux->sig->fecha = aux->fecha;

strcpy( aux->vendedor, tmpV );
aux->total = tmpT;

```

```

aux->fecha = tmpF;

fin=false;
}
aux=aux->sig;
}
}
else{
// descendente
while (aux->sig!=NULL || fin==false){
if (aux->sig==NULL)
{
fin=true;
aux=ventas;
}
if(aux->fecha < aux->sig->fecha){
// guardamos en una variable temporal
strcpy( tmpV, aux->sig->vendedor );
tmpT = aux->sig->total;
tmpF = aux->sig->fecha;

// cambiamos los datos
strcpy( aux->sig->vendedor, aux->vendedor );
aux->sig->total = aux->total;
aux->sig->fecha = aux->fecha;

strcpy( aux->vendedor, tmpV );
aux->total = tmpT;
aux->fecha = tmpF;

fin=false;
}
aux=aux->sig;
}
}
}

```

MAIN.CPP

```

#include <stdio.h>
#include <iostream>
#include <string.h>
#include "Registro.cpp"

int main(void){
Registro registro = Registro();
int opc, total, fecha, tmp=0, o1=0, o2=0;
char vendedor[25];

```

```

printf("\t BIENVENIDO UWU");
do
{
printf("\n\n1.-Registrar venta \n2.-eliminar venta \n3.-ordenar\n4.-salir\n");
scanf("%d",&opc);
switch(opc){
case 1:
printf("\n\ningresa el nombre del vendedor: ");
scanf("%24s",vendedor);
printf("ingresa el total vendido: ");
scanf("%d",&total);
printf("ingresar la fecha: ");
scanf("%d",&fecha);
registro.registrar(vendedor, total,fecha);
break;
case 2:
registro.mostrar();
printf("\n\ningresar el numero del registro que deseas eliminar");
scanf("%d",&tmp);
registro.eliminar(tmp);
printf("\n");
break;
case 3:
do
{
printf("\n\n1.-ordenar por vendedor\n2.-ordenar por total\n3.-ordenar por fecha\n");
scanf("%d",&o1);
} while (o1<1 || o1>3);
do
{
printf("\n\n1.-ascendente\n2.-descendente\n");
scanf("%d",&o2);
} while (o2<1 || o2>2);
if (o1==1)
{
registro.ordenarVendedor(o2);
registro.mostrar();
}
if (o1==2)
{
registro.ordenarTotal(o2);
registro.mostrar();
}
if (o1==3)
{
registro.ordenarFecha(o2);
registro.mostrar();
}
break;

```



```

}
} while (opc!=4);
return 0;
}

```

CAPTURAS

Visual Studio Code interface showing the execution of a C++ program. The terminal output is as follows:

```

irvyn@irvyn-System-Product-Name:~/Documentos/SO1/practical1$ ./main
BIENVENIDO UUU

1.-Registrar venta
2.-eliminar venta
3.-ordenar
4.-salir
1

ingresa el nombre del vendedor: jose
ingresa el total vendido: 10
ingresar la fecha: 31

```

Visual Studio Code interface showing the execution of a C++ program. The terminal output is as follows:

```

1.-Registrar venta
2.-eliminar venta
3.-ordenar
4.-salir
1

ingresa el nombre del vendedor: irvyn
ingresa el total vendido: 25
ingresar la fecha: 64

1.-Registrar venta
2.-eliminar venta
3.-ordenar
4.-salir
3

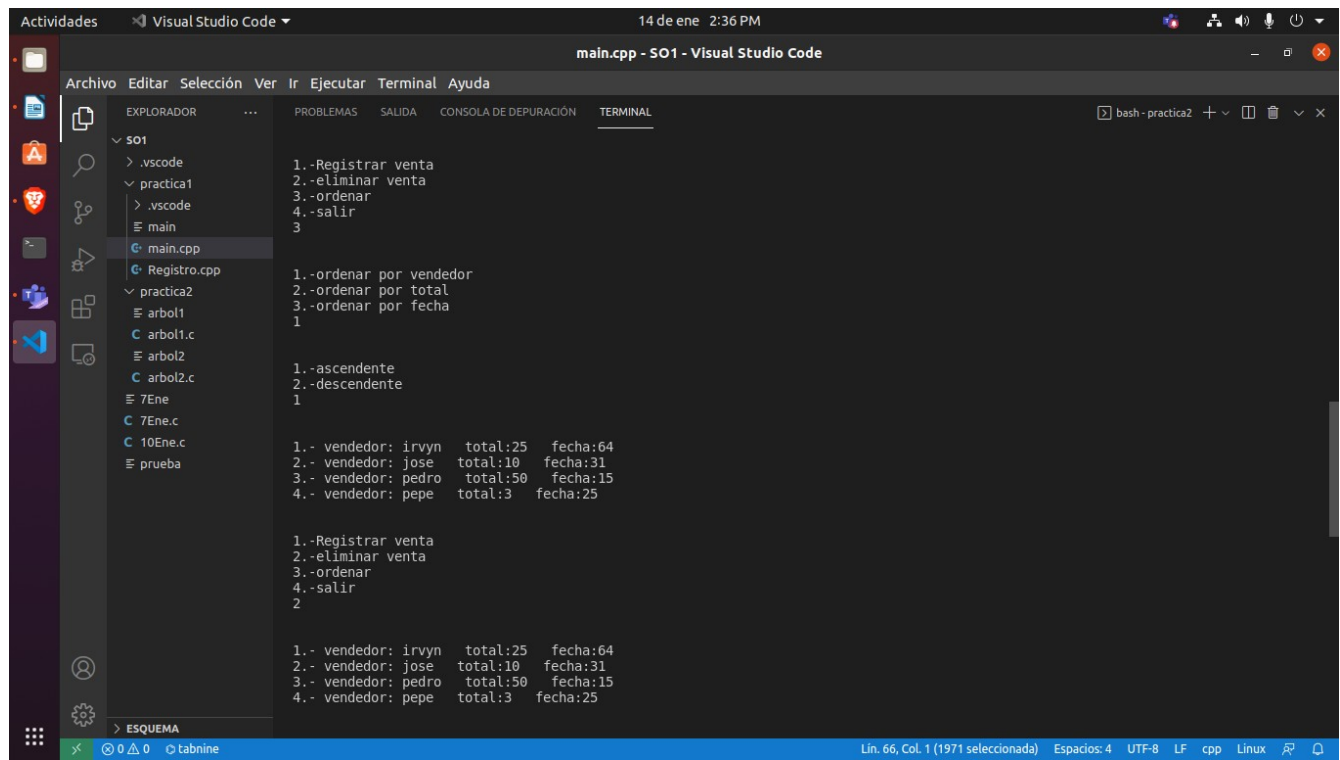
1.-ordenar por vendedor
2.-ordenar por total
3.-ordenar por fecha
2

1.-ascendente
2.-descendente
2

1.- vendedor: pedro    total:50    fecha:15
2.- vendedor: irvyn    total:25    fecha:64
3.- vendedor: jose     total:10    fecha:31
4.- vendedor: pepe     total:3     fecha:25

1.-Registrar venta

```



Actividades Visual Studio Code 14 de ene 2:36 PM

main.cpp - S01 - Visual Studio Code

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

EXPLORADOR

- S01
 - .vscode
 - practica1
 - .vscode
 - main
 - main.cpp
 - Registro.cpp
 - practica2
 - arbol1
 - arbol1.c
 - arbol2
 - arbol2.c
 - 7Ene
 - 7Ene.c
 - 10Ene.c
 - prueba

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

bash - practica2

```
2.- vendedor: jose total:10 fecha:31
3.- vendedor: pedro total:50 fecha:15
4.- vendedor: pepe total:3 fecha:25

ingresar el numero del registro que deseas eliminar1

1.-Registrar venta
2.-eliminar venta
3.-ordenar
4.-salir
3

1.-ordenar por vendedor
2.-ordenar por total
3.-ordenar por fecha
3

1.-ascendente
2.-descendente
1

1.- vendedor: pedro total:50 fecha:15
2.- vendedor: pepe total:3 fecha:25
3.- vendedor: jose total:10 fecha:31

1.-Registrar venta
2.-eliminar venta
3.-ordenar
4.-salir
4
```

ESQUEMA

irvyn@irvyn-System-Product-Name:~/Documentos/S01/practica1\$ cd ..

Lin. 66, Col. 1 (1971 seleccionada) Espacios: 4 UTF-8 LF cpp Linux