

Entregable 09.Implementación de interfaces de usuario II

```
#ifndef MENUAGENTES_H_INCLUDED
#define MENUAGENTES_H_INCLUDED

#include <iostream>
#include <string>

#include "listaagentes.h"
#include "menucliente.h"
#include "listexception.h"
#include "nombre.h"
#include "horario.h"

class MenuAgentes{
private:
    ListaAgentes* listaDeAgentes;
    MenuCliente menuClientes;
public:
    MenuAgentes();
    MenuAgentes(ListaAgentes*);

    void menuPrincipal();
    void agregarAgente();
    void borrarAgente();
    void modificarAgente();
    void modificarListaDeClientesPorAgente();
    void ordenar();
    void mostrarListaDeAgentes();
    void mostrarListaDeClientesPorAgente();
    void borrarLista();

};

#endif // MENUAGENTES_H_INCLUDED

#include "menuagentes.h"

using namespace std;

MenuAgentes::MenuAgentes(): listaDeAgentes(nullptr) {

}
```

```
MenuAgentes::MenuAgentes(ListaAgentes* l) {  
    listaDeAgentes = l;  
}
```

```
void MenuAgentes::menuPrincipal() {  
    char op;  
    char op2;  
  
    do {  
  
        cout << "Sistema de administracion call-center" << endl;  
        cout << "1) Agregar Agente" << endl;  
        cout << "2) Borrar Agente" << endl;  
        cout << "3) Modificar Agente" << endl;  
        cout << "4) Ordenar la Lista" << endl;  
        cout << "5) Mostrar Lista" << endl;  
        cout << "6) Borrar Lista" << endl;  
        cout << "7) Salir" << endl << endl;  
  
        cout << "Elija una opcion: ";  
        cin >> op;  
        cin.ignore();  
        switch(op) {  
  
            case '1':  
                cout << "Agregar Agente:" << endl;  
                agregarAgente();  
                break;  
            case '2':  
                cout << "Borrar Agente:" << endl;  
                borrarAgente();  
                break;  
            case '3':  
                cout << "Modificar Agente:" << endl;  
                do {  
                    cout << "1) Modificar Agente" << endl;  
                    cout << "2) Modificar Lista de Clientes Del Agente" << endl;  
                    cout << "3) Salir" << endl;  
                    cin >> op2;  
                    cin.ignore();  
                    switch(op2) {  
                        case '1':  
                            cout << "Modificar Agente:" << endl;  
                            modificarAgente();  
                            break;  
                        case '2':  
                            cout << "Modificar Lista de Clientes Del Agente:" << endl;  

```

```

        modificarListaDeClientesPorAgente();
        break;
    case '3':
        cout<<"Salir"<<endl;
        break;
    default:
        cout<<"Opcion invalida"<<endl;
    }
} while(op2 != '3');
break;
case '4':
    cout <<"Ordenar la Lista:"<<endl;
    ordenar();
    break;
case '5':
    do {
        cout <<"Mostrar Lista:"<< endl;
        cout<<"1)Mostrar Lista de Agentes"<<endl;
        cout<<"2)Mostrar Lista de Clientes por Agente"<<endl;
        cout<<"3)Salir"<<endl;
        cin>>op2;
        cin.ignore();
        switch(op2) {
            case '1':
                cout<<"Mostrar Lista de Agentes:"<<endl;
                mostrarListaDeAgentes();
                break;
            case '2':
                cout<<"Mostrar Lista de Clientes por Agente:"<<endl;
                mostrarListaDeClientesPorAgente();
                break;
            case '3':
                cout<<"Salir"<<endl;
                break;
            default:
                cout<<"Opcion Invalida"<<endl;
        }
    } while(op2 != '3');

    break;
case '6':
    cout <<"Borrar Lista"<<endl;
    borrarLista();
    break;
case '7':
    cout <<"Salir" <<endl;
    getchar();

```

```

        break;

    default:
        cout << "Opcion invalida" << endl;
        cout << "presione enter para continuar" << endl;
        getchar();
    }
} while (op != '7');

}

void MenuAgentes::agregarAgente() {
    char op;
    Agente miAgente;
    Nombre miNombre;
    Horario miHorario;
    string cadena;
    int entero;

    cout << "Ingresa el nombre del agente: (apellidos y luego nombre)" << endl;
    getline(cin, cadena, ' ');
    miNombre.setApellido(cadena);
    getline(cin, cadena, ' ');
    miNombre.setApellido(cadena);
    getline(cin, cadena, '\n');
    miNombre.setNombre(cadena);
    miAgente.setNombre(miNombre);

    cout << "Ingresa el horario de atencion:" << endl;
    cout << "Hora:";
    cin >> entero;
    miHorario.setHora(entero);
    cin.ignore();
    cout << endl;
    cout << "Minuto:";
    cin >> entero;
    cin.ignore();
    miHorario.setMinuto(entero);
    miAgente.setHorarioDeAtencion(miHorario);
    cout << endl;

    cout << "Ingresa el numero de extencion:" << endl;
    cin >> entero;
    cin.ignore();
    miAgente.setNumeroDeExtencion(entero);

    cout << "Ingresa las horas extras:" << endl;

```

```

    cin>>entero;
    cin.ignore();
    miAgente.setHorasExtras(entero);

    cout<<"Ingresa la especialidad"<<endl;
    getline(cin,cadena,'\n');
    miAgente.setEspecialidad(cadena);

    listaDeAgentes->insertar(miAgente);

    cout<<"Quieres ingresar la lista de clientes? (s/n)"<<endl;
    cin>>op;
    cin.ignore();
    if(op == 's' or op == 'S') {
        menuClientes= listaDeAgentes->ultimoNodo()->getListaDeClientes();
        menuClientes.menuPrincipal();
    }
}

void MenuAgentes::borrarAgente() {
    Agente miAgente;
    Nombre miNombre;
    string cadena;

    cout<<"Ingresa el nombre del agente: (apellidos y luego nombre)"<<endl;
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,'\n');
    miNombre.setNombre(cadena);
    miAgente.setNombre(miNombre);

    try {
        listaDeAgentes->eliminar(miAgente);
    } catch(ListException) {
        cout<<"List Exception, eliminar"<<endl;
    }
}

void MenuAgentes::modificarAgente() {
    Agente miAgente;
    Nombre miNombre;
    Horario miHorario;
    string cadena;

```

```
int entero;
```

```
cout<<"Ingresa el nombre del agente: (apellidos y luego nombre)"<<endl;
getline(cin,cadena,' ');
miNombre.setApellido(cadena);
getline(cin,cadena,' ');
miNombre.setApellido(cadena);
getline(cin,cadena,'\n');
miNombre.setNombre(cadena);
miAgente.setNombre(miNombre);
```

```
try {
    NodoAgente* aux = listaDeAgentes->localiza(miAgente);
    cout<<aux->toString()<<endl<<endl;
```

```
    cout<<"Ingresa el nombre del agente: (apellidos y luego nombre)"<<endl;
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,'\n');
    miNombre.setNombre(cadena);
    miAgente.setNombre(miNombre);
```

```
    cout<<"Ingresa el horario de atencion:"<<endl;
    cout<<"Hora:";
    cin>>entero;
    miHorario.setHora(entero);
    cout<<endl;
    cout<<"Minuto:";
    cin>>entero;
    miHorario.setMinuto(entero);
    miAgente.setHorarioDeAtencion(miHorario);
    cout<<endl;
```

```
    cout<<"Ingresa el numero de extencion:"<<endl;
    cin>>entero;
    miAgente.setNumeroDeExtencion(entero);
```

```
    cout<<"Ingresa las horas extras:"<<endl;
    cin>>entero;
    miAgente.setHorasExtras(entero);
```

```
    cout<<"Ingresa la especialidad"<<endl;
    getline(cin,cadena,'\n');
    miAgente.setEspecialidad(cadena);
```

```

        aux->setAgente(miAgente);
    } catch(ListException) {
        cout<<"List Exception, modificar agente"<<endl;
    }
}

void MenuAgentes::modificarListaDeClientesPorAgente() {
    Agente miAgente;
    Nombre miNombre;
    Horario miHorario;
    string cadena;
    int entero;

    cout<<"Ingresa el nombre del agente: (apellidos y luego nombre)"<<endl;
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,'\n');
    miNombre.setNombre(cadena);
    miAgente.setNombre(miNombre);

    cout<<"Ingresa el horario de atencion:"<<endl;
    cout<<"Hora:";
    cin>>entero;
    miHorario.setHora(entero);
    cout<<endl;
    cout<<"Minuto:";
    cin>>entero;
    miHorario.setMinuto(entero);
    miAgente.setHorarioDeAtencion(miHorario);
    cout<<endl;
    try {
        NodoAgente *aux = listaDeAgentes->localiza(miAgente);
        menuClientes = aux->getListaDeClientes();
        menuClientes.menuPrincipal();
    } catch(ListException) {
        cout<<"List Exception, mostrar lista de clientes por agente"<<endl;
    }
}

void MenuAgentes::ordenar() {
    try {
        listaDeAgentes->ordena();
    } catch(ListException) {

```

```

        cout<<"List Exception, ordenar"<<endl;
    }
}

void MenuAgentes::mostrarListaDeAgentes() {
    listaDeAgentes->toString();
}

void MenuAgentes::mostrarListaDeClientesPorAgente() {
    Agente miAgente;
    Nombre miNombre;
    Horario miHorario;
    string cadena;
    int entero;

    cout<<"Ingresa el nombre del agente: (apellidos y luego nombre)"<<endl;
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,' ');
    miNombre.setApellido(cadena);
    getline(cin,cadena,'\n');
    miNombre.setNombre(cadena);
    miAgente.setNombre(miNombre);

    cout<<"Ingresa el horario de atencion:"<<endl;
    cout<<"Hora:";
    cin>>entero;
    miHorario.setHora(entero);
    cout<<endl;
    cout<<"Minuto:";
    cin>>entero;
    miHorario.setMinuto(entero);
    miAgente.setHorarioDeAtencion(miHorario);
    cout<<endl;

    try {
        NodoAgente* aux = listaDeAgentes->localiza(miAgente);
        menuClientes = aux->getListaDeClientes();
        menuClientes.mostrarLista();
    } catch(ListException) {
        cout<<"List Exception, mostrar lista de clientes por agente"<<endl;
    }
}

void MenuAgentes::borrarLista() {
    try {
        listaDeAgentes->eliminarTodo();
    }
}

```



```
    } catch(ListException) {
        cout<<"List Exception, borrar todo"<<endl;
    }
}
```

Se le inyecta texto con las opciones 1agregar agente,s para agregar una lista al agente,1agregar cliente 3 veces y mostrar la lista a través del agente.