

Alberto Garcia Ramirez

216100773

```
#ifndef NOMBRE_H_INCLUDED
#define NOMBRE_H_INCLUDED

#include <string>

class Nombre {
private:
    std::string nombre;
    std::string apellido;
public:
    Nombre();
    Nombre(const Nombre&);

    std::string getNombre();
    std::string getApellido();

    std::string toString() const;

    void setNombre(const std::string&);
    void setApellido(const std::string&);

    Nombre& operator = (const Nombre&);
    bool operator == (const Nombre&);
    bool operator < (const Nombre&);
    bool operator <= (const Nombre&);
    bool operator > (const Nombre&);
    bool operator >= (const Nombre&);

    friend std::ostream& operator<<(std::ostream&,Nombre&);
    friend std::istream& operator>>(std::istream&,Nombre&);
};

#endif // NOMBRE_H_INCLUDED

#ifndef HORARIO_H_INCLUDED
#define HORARIO_H_INCLUDED

#include <string>
#include <chrono>
#include <time.h>
#include <cstdio>
#include <cstdlib>
```

```

class Horario{
private:
    int hora;
    int minuto;

    int toInt()const ;

public:
    Horario();
    Horario(const Horario&);

    int getHora();
    int getMinuto();

    std::string toString();

    void setHora(const int&);
    void setMinuto(const int&);

    Horario& operator = (const Horario&);
    bool operator ==(const Horario&);
    bool operator <(const Horario&);
    bool operator <=(const Horario&);
    bool operator >(const Horario&);
    bool operator >=(const Horario&);

    friend std::ostream& operator<<(std::ostream&,Horario&);
    friend std::istream& operator>>(std::istream&,Horario&);
};

```

```

#endif // HORARIO_H_INCLUDED

```

```

#ifndef DURACION_H_INCLUDED
#define DURACION_H_INCLUDED

```

```

#include <cstdio>
#include <string>
#include <iostream>

```

```

class Duracion{
private:
    int hora;
    int minuto;
    int segundo;
public:

```

```

Duracion();
Duracion(const Duracion&);

int getHora();
int getMinuto();
int getSegundo();

std::string toString();

void setHora(const int&);
void setMinuto(const int&);
void setSegundo(const int&);

Duracion& operator = (const Duracion&);

friend std::ostream& operator<<(std::ostream&,Duracion&);
friend std::istream& operator>>(std::istream&,Duracion&);
};

#endif // DURACION_H_INCLUDED

#ifndef CLIENTE_H_INCLUDED
#define CLIENTE_H_INCLUDED

#include <string>
#include <iostream>

#include "nombre.h"
#include "horario.h"
#include "duracion.h"

class Cliente{
private:
    Nombre nombre;
    Horario horaDeLlamada;
    Duracion duracion;
public:
    Cliente();
    Cliente(const Cliente&);

    std::string getNombre();
    std::string gethoraDeLlamada();
    std::string getDuracion();

    std::string toString();

```

```

void setNombre(const Nombre&);
void setHoraDeLlamada(const Horario&);
void setDuracion(const Duracion&);

Cliente& operator = (const Cliente&);
bool operator == (const Cliente&);
bool operator < (const Cliente&);
bool operator <= (const Cliente&);
bool operator > (const Cliente&);
bool operator >= (const Cliente&);

friend std::ostream& operator<<(std::ostream&, Cliente&);
friend std::istream& operator>>(std::istream&, Cliente&);
};

#endif // CLIENTE_H_INCLUDED

#ifndef AGENTE_H_INCLUDED
#define AGENTE_H_INCLUDED

#include <string>
#include <cstdio>
#include <cstdlib>
#include <iostream>

#include "nombre.h"
#include "horario.h"
#include "listacliente.h"

class Agente{
private:
    Nombre miNombre;
    Horario horarioEntrada;
    Horario horarioSalida;
    int numeroDeExtencion;
    int horasExtras;
    std::string especialidad;

    ListaCliente miListaDeClientes;
public:
    Agente();
    Agente(const Agente&);

    std::string getNombre();
    std::string getHorarioEntrada();

```

```

std::string getHorarioSalida();
std::string getNumeroDeExtencion();
std::string getHorasExtras();
std::string getEspecialidad();
ListaCliente* getListaDeClientes();

std::string toString();

void setNombre(const Nombre&);
void setHorarioEntrada(const Horario&);
void setHorarioSalida(const Horario&);
void setNumeroDeExtencion(const int&);
void setHorasExtras(const int&);
void setEspecialidad(const std::string&);
void setListaDeClientes(ListaCliente&);

Agente& operator = (const Agente&);
bool operator == (const Agente&);
bool operator < (const Agente&);
bool operator <= (const Agente&);
bool operator > (const Agente&);
bool operator >= (const Agente&);

friend std::ostream& operator<<(std::ostream&, Agente&);
friend std::istream& operator>>(std::istream&, Agente&);
};

#endif // AGENTE_H_INCLUDED

#ifndef NODOCLIENTE_H_INCLUDED
#define NODOCLIENTE_H_INCLUDED

#include <string>

#include "cliente.h"

class NodoCliente{
private:
    Cliente cliente;
    NodoCliente *siguiente;
public:
    NodoCliente();
    NodoCliente(const Cliente&);

    Cliente& getCliente();
    NodoCliente *getSiguiente();

```

```

    std::string toString();

    void setCliente(const Cliente&);
    void setSiguiente(NodoCliente *);
};

#endif // NODOCLIENTE_H_INCLUDED

#ifndef NODOAGENTE_H_INCLUDED
#define NODOAGENTE_H_INCLUDED

#include <string>

#include "agente.h"
class NodoAgente{
private:
    Agente miAgente;
    NodoAgente *siguiente;
    NodoAgente *anterior;
public:
    NodoAgente();
    NodoAgente(const Agente&);

    Agente& getAgente();
    NodoAgente *getSiguiente();
    NodoAgente *getAnterior();

    std::string toString();

    void setAgente(const Agente&);
    void setSiguiente(NodoAgente*);
    void setAnterior(NodoAgente*);

    NodoAgente& operator = (const NodoAgente&);
};

#endif // NODOAGENTE_H_INCLUDED

#ifndef LISTACLIENTE_H_INCLUDED
#define LISTACLIENTE_H_INCLUDED

#include <string>
#include <fstream>

```

```
#include "nodocliente.h"
#include "cliente.h"
#include "listexception.h"
```

```
class ListaCliente{
private:
    NodoCliente* ultimo;
    NodoCliente* primerInsertado;
    NodoCliente* auxiliar;
    NodoCliente* auxiliar2;

    void intercambiar(NodoCliente*,NodoCliente*);
    void copiarTodo(const ListaCliente&);
public:
    ListaCliente();
    ListaCliente(const ListaCliente&);
    ~ListaCliente();

    bool isEmpty();
    void insertar(const Cliente&);
    void eleminar(const Cliente&);
    NodoCliente* primerNodo();
    NodoCliente* ultimoNodo();
    NodoCliente* anterior(NodoCliente*);
    NodoCliente* siguiente(NodoCliente*);
    NodoCliente* localiza(const Cliente&);
    void ordena();
    void ordena(NodoCliente*,NodoCliente*);
    std::string recupera(const Cliente&);
    std::string toString();
    void guardarAlDisco(const std::string&);
    void leerDelDisco(const std::string&);
    void eliminarTodo();

    ListaCliente& operator = (const ListaCliente&);
};
```

```
#endif // LISTACLIENTE_H_INCLUDED
```

```
#ifndef LISTAAGENTES_H_INCLUDED
#define LISTAAGENTES_H_INCLUDED
```

```
#include <string>
#include <iostream>
#include <fstream>
```

```

#include "nodoagente.h"
#include "agente.h"
#include "listexception.h"
#include "menucliente.h"
class ListaAgentes{
private:
    NodoAgente *ultimoInsertado;
    NodoAgente *primerInsertado;
    NodoAgente *auxiliar1;
    NodoAgente *auxiliar2;

    void intercambiar(NodoAgente*,NodoAgente*);
    void copiarTodo(const ListaAgentes&);
public:
    ListaAgentes();
    ListaAgentes(const ListaAgentes&);
    ~ListaAgentes();

    bool isEmpty();
    void insertar(const Agente&);
    void eliminar(const Agente&);
    NodoAgente* primerNodo();
    NodoAgente* ultimoNodo();
    NodoAgente* anterior(NodoAgente*);
    NodoAgente* siguiente(NodoAgente*);
    NodoAgente* localiza(const Agente&);
    void ordenaPorNombre();
    void ordenaPorNombre(NodoAgente*,NodoAgente*);
    void ordenaPorEspecialidad();
    void ordenaPorEspecialidad(NodoAgente*,NodoAgente*);
    std::string recupera(const Agente&);
    std::string toString();
    std::string toStringAgentes();
    void guardarAlDisco(const std::string&);
    void leerDelDisco(const std::string&);
    void eliminarTodo();

    ListaAgentes& operator = (const ListaAgentes&);
};

#endif // LISTAAGENTES_H_INCLUDED

#ifdef MENUCLIENTE_H_INCLUDED
#define MENUCLIENTE_H_INCLUDED

```



```
#include <iostream>
#include <string>
```

```
#include "duracion.h"
#include "cliente.h"
#include "nombre.h"
#include "nodocliente.h"
#include "listacliente.h"
```

```
class MenuCliente{
private:
    ListaCliente* MiLista;
public:
    MenuCliente();
    MenuCliente(ListaCliente*);

    void menuPrincipal();
    void agregarCliente();
    void borrarCliente();
    void modificarCliente();
    void ordenarLista();
    void mostrarLista();
    void guardarAlDisco();
    void leerDelDisco();
    void borrarLista();
};
```

```
#endif // MENUCLIENTE_H_INCLUDED
```

```
#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 ordenarPorNombre();  
void ordenarPorEspecialidad();  
void mostrarListaDeAgentes();  
void mostrarListaDeClientesPorAgente();  
void guardarAlDisco();  
void leerDelDisco();  
void borrarLista();
```

```
};
```

```
#endif // MENUAGENTES_H_INCLUDED
```