

```

/* Programa que implementa los métodos necesarios para
manejar la estructura de datos cola. El programa
implementa métodos para Ingresar y retirar datos de
la cola, mostrar el primero y el último, al igual
que mostrar todos los elementos de la cola. Los datos
que se pueden almacenar en cada uno son de tipo char.
Programa elaborado por: Edgar Romero Rodríguez,
                        Ing. De Sistemas - UIS
*/
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <string>

// Implementación de la clase colas

using namespace std;

class colas
{
    /* Se define la estructura del nodo de la lista,
    en este caso se guardarán datos de tipo entero.
    */
    struct nodo {
        char nomb[10];
        char apel[10];
        nodo *siguiente;
    };
    // Definición de variables para controlar la cola

    nodo *Ini_cola, *Fin_cola, *Aux, *Aux1;

    /* Ini_cola es puntero que apunta al primer elemento de la cola
    Fin_cola es un puntero que apunta al último elemento de la cola
    Aux y Aux1 son variables auxiliares para recorrer la cola
    */

public:
    colas(); // Constructor
    void Insertar(char nom[10], char ape[10]);
    void QuitarNodo();
    int ConsultarPosicion(char nom[10]);
    void MostrarDatos();
    void VerPrimero();
    ~colas(); // Destructor
}; // Finaliza la definición de métodos...

// Área donde se implementan los métodos

// Constructor
colas::colas()
{
    Ini_cola = NULL;
    Fin_cola = NULL;
}

// Método para insertar un nodo en la cola (se inserta después del último)

```

```

void colas::Insertar(char nom[10], char ape[10])
{
    //Se crea un nuevo nodo usando la variable auxiliar
    Aux = new nodo;
    if (Aux==NULL) // Se verifica si hay memoria disponible
    {
        cout<<"No hay memoria suficiente para crear mas nodos...";
        system("PAUSE");
    }
    // Se copian los datos leidos en el nodo de la cola
    strcpy(Aux->nomb, nom);
    strcpy(Aux->apel, ape);
    // Como este pasa a ser el ultimo nodo, se pone apuntando a NULL
    Aux->siguiente = NULL;
    if (IniCola == NULL)
    {
        IniCola = Aux;
    }
    else
    {
        // Enlazo el nuevo nodo con el que estaba de ultimo (FinCola apunta al ultimo)
        Aux1 = FinCola;
        Aux1->siguiente = Aux;
        // El nuevo nodo ahora sera el ultimo, por lo tanto FinCola apunta a el.
    }
    FinCola = Aux;
} // Fin del metodo Insertar

//Metodo para retirar el primer elemento de la lista
void colas::QuitarNodo(){
    if(IniCola!=NULL) {
        Aux = IniCola;
        cout<<"Se retira: "<<Aux-> nomb<<"", "<<Aux-> apel<<endl<<endl;
        IniCola = Aux->siguiente;
        delete Aux;
    } else {cout<<"No hay elementos en cola..."<<endl<<endl;};
} // Fin del metodo

// Metodo para consultar la posicion de un nodo en la cola
int colas::ConsultarPosicion(char nom[10])
{
    int pos = 0;
    bool seguir = false; //Indica cuando se ha encontrado el dato
    bool hallado = false; //Se pone en true si se hallo el nombre
    Aux = IniCola;
    cout<<"\nFavor esperar...buscando informacion...\n"<<endl;
    // Se recorre la cola para buscar si los datos coinciden
    if (Aux!=NULL) seguir = true;
    while(seguir)
    {
        //Se compara el contenido del nodo con el dato buscado
        if((strcmp(Aux->nomb, nom)==0)){
            seguir = false; // dato encontrado.
            hallado = true;
        }
        pos++; //Incremento en uno el contador de posicion
        Aux = Aux->siguiente; // paso al siguiente nodo
        if(Aux==NULL) seguir = false;
    }
}

```

```

    }
    if (hallado) return pos; else return 0;
}

void colas::MostrarDatos()
{
    Aux = IniCola;
    cout<<"\nINFORMACION DE LA COLA"<<endl;
    cout<<endl<<"Inicio: "<<endl;
    while(Aux!=NULL)
    {
        //Se imprime el contenido de cada nodo (campo nomb)
        cout<<"    "<<Aux->nomb<<"", "<<Aux->apel<<endl;
        Aux = Aux->siguiente;
    }
    cout<<"Final!!! "<<endl<<endl;
} // Final del metodo mostrar la lista

//Metodo para mostrar el primero de la cola
void colas::VerPrimero(){
    cout<<"\nEl primero en cola es: "<<IniCola->nomb<<"", "<<IniCola->
    >apel<<endl<<endl;
} // Fin del metodo

//Destructor
colas::~~colas()
{
    Aux = IniCola;
    while(Aux!=NULL)
    {
        //Se imprime el contenido de cada nodo (campo nomb)
        Aux1 = Aux;
        delete Aux;
        Aux = Aux1->siguiente;
    }
    delete IniCola;
    delete FinCola;
    delete Aux1;
    cout<<"...la cola se elimino de la memoria RAM..."<<endl<<endl;
    system("PAUSE");
} // fin de la implementacion de la clase colas

//Este es el programa principal..
main()
{
    int opc;
    colas DatosCola;
    do
    {
        system("cls"); // Menu de opciones disponibles
        cout<<"*** OPCIONES DISPONIBLES ***\n"<<endl;
        cout<<"1. Incluir nodo"<<endl;
        cout<<"2. Borrar un nodo"<<endl;
        cout<<"3. Listar datos"<<endl;
        cout<<"4. Consultar primero"<<endl;
        cout<<"5. Consultar posicion"<<endl;
        cout<<"6. Terminar"<<endl;
        cout<<"digite su opcion: ";
        //Lectura de la opcion elegida por el usuario
    }
}

```

```

do {
    cin>>opc;
} while (opc<1 ||opc>6);
switch(opc)
{
    case 1:
    {
        char nombre[10], apellido[10];
        cout<<"Digite el Nombre: ";
        cin>>nombre;
        cout<<"Digite el apellido: ";
        cin>>apellido;
        DatosCola.Insertar(nombre, apellido);
        break;
    }
    case 2:
    {
        DatosCola.QuitarNodo();
        break;
    }
    case 3:
    {
        DatosCola.MostrarDatos();
        system("PAUSE");
        break;
    }
    case 4:
    {
        DatosCola.VerPrimero();
        system("PAUSE");
        break;
    }
    case 5:
    {
        char nombre[10];
        int rta;
        cout<<"\nNombre a buscar: ";
        cin>>nombre;
        rta = DatosCola.ConsultarPosicion(nombre);
        if (rta=0) cout<<"\nEl nombre buscado> "<<nombre<<" , no se enocntro";
        else cout<<nombre<<" , ocupa la posicion: "<<rta<<endl;
        system("PAUSE");
        break;
    }
    } // Fin del switch
} while (opc <=5);
cout<<"Programa terminado "<<endl;
system("PAUSE");
return EXIT_SUCCESS;
} // Fin del programa principal

```