

Clase Principal

```
using System;
namespace Practica9
{
    class Program
    {
        static void Main(string[] args)
        {
            Cola c = new Cola();
            int OpcMenu = 0;
            do
            {
                Console.WriteLine("Cola");
                Console.WriteLine(" 1. Ingresar dato");
                Console.WriteLine(" 2. Desplegar cola");
                Console.WriteLine(" 3. Eliminar dato");
                Console.WriteLine("Elija opcion: ");
                OpcMenu = int.Parse(Console.ReadLine());
                switch (OpcMenu)
                {
                    case 1:
                        Console.WriteLine("\n Ingrese dato");
                        c.InsertarNodo();
                        c.InsertarNodo();
                        c.InsertarNodo();
                        c.InsertarNodo();
                        c.InsertarNodo();
                        break;
                    case 2:
                        Console.WriteLine("\n Desplegar cola");
                        c.DesplegarCola();
                        break;
                    case 3:
                        Console.WriteLine("\n Eliminar dato");
                        c.EliminarNodo();
                        break;
                    default:
                        Console.WriteLine("Opcion invalida");
                        break;
                }
            }
            while (OpcMenu != 0);
        }
    }
}
```

Clase Nodo

```
namespace Practica9
{
    public class Nodo
    {
        private int dato;
        private Nodo siguiente;
        public int Dato
        {
            get { return dato; }
            set { dato = value; }
        }
        public Nodo Siguiente
        {
            get { return siguiente; }
            set { siguiente = value; }
        }
    }
}
```

Clase Cola

```
using System;

namespace Practica9
{
    public class Cola
    {
        private Nodo Primero = new Nodo();
        private Nodo Ultimo = new Nodo();
        public Cola()
        {
            Primero = null;
            Ultimo = null;
        }

        public void InsertarNodo()
        {
            Nodo Nuevo = new Nodo(); ;
            Console.WriteLine("\n Ingrese dato: ");
            Nuevo.Dato = int.Parse(Console.ReadLine());
            if (Primero == null)
            {
                Primero = Nuevo;
            }
        }
    }
}
```

```

        Primero.Siguiente = null;
        Ultimo = Nuevo;
    }
    else
    {
        Ultimo.Siguiente = Nuevo;
        Nuevo.Siguiente = null;
        Ultimo = Nuevo;
    }
}

```

```

public void DesplegarCola()
{
    Nodo Actual = new Nodo();
    Actual = Primero;
    if (Primero != null)
    {
        while (Actual != null)
        {
            Console.WriteLine(" " + Actual.Dato);
            Actual = Actual.Siguiente;
        }
    }
    else
    {
        Console.WriteLine("La cola se encuentra vacia");
    }
}

```

```

public void EliminarNodo()
{
    Nodo Actual = new Nodo();
    Nodo Anterior = new Nodo();
    Anterior = null;
    Actual = Primero;
    bool Encontrado = false;
    Console.WriteLine("Ingrese el dato a eliminar: ");
    int NodoBuscar = int.Parse(Console.ReadLine());

    if (Primero != null && Encontrado != true)
    {
        while (Actual != null)
        {
            if (Actual.Dato == NodoBuscar)
            {
                Console.WriteLine("El dato ({0}) a sido encontrado", NodoBuscar);
            }
        }
    }
}

```

```

        if (Actual == Primero)
        {
            Primero = Primero.Siguiente;
        }
        else if (Actual == Ultimo)
        {
            Anterior.Siguiente = null;
            Ultimo = Anterior;
        }
        else
        {
            Anterior.Siguiente = Actual.Siguiente;
        }
        Console.WriteLine("\n Nodo eliminado \n");
        Encontrado = true;
    }
    Anterior = Actual;
    Actual = Actual.Siguiente;
}
if (!Encontrado)
{
    Console.WriteLine("\n Dato no encontrado \n");
}
}
else
{
    Console.WriteLine("La cola se encuentra vacia");
}
}
}
}

```