

1.- ESPECIFICACIÓN:

tad Cola(telemento) { Cola está formada por elementos de tipo telemento }

Especificación de las operaciones:

```
accion iniciarCola(sal Cola C)
{ Inicia C como una cola vacía }

accion añadir(e/s Cola C, ent telemento d)
{ Añade a C el elemento d }

funcion colaVacía( Cola C ) devuelve booleano
{ Si C está vacía devuelve VERDAD y FALSO en otro caso }

funcion primero( Cola C ) devuelve telemento
{ Devuelve el elemento más antiguo de la cola C y no
  modifica la cola }

accion eliminar( e/s Cola C )
{ Modifica la cola C eliminando el elemento más
  antiguo }
```

2a.- IMPLEMENTACIÓN ESTÁTICA:

• **REPRESENTACIÓN:**

```
Tipodef telemento=tVector[MAXCOLA]
constante
    MAXCOLA=....; {representa el tamaño máximo de la cola}

tipo
    Cola = registro
        tvector datos
        entero primero, ultimo
        entero num
    freg
```

• **INTERPRETACIÓN:**

num representa el número de elementos de la cola, los cuales se encuentran almacenados entre las componentes primero y ultimo, estando en la componente primero el elemento más antiguo de la cola y en último el más reciente.

•

IMPLEMENTACIÓN DE LAS OPERACIONES:

```
accion iniciarCola( sal Cola C )
{ Inicia C como una cola vacía }
principio
    C.num ← 0
    C.primer0 ← 0
    C.ultimo ← -1
fin

accion añadir( e/s Cola C, ent telemento d)
{ Añade a C el elemento d }

principio
    si C.num<MAXCOLA entonces
        C.num ← C.num + 1
        C.ultimo ← suma_uno(C.ultimo)
        C.datos[C.ultimo] ← d
    fsi
fin

funcion colaVacia( Cola C ) devuelve booleano
{ Si C está vacía devuelve VERDAD y FALSO en otro caso }

principio
    devuelve( C.num = 0 )
fin

funcion primero( Cola C ) devuelve telemento
{ Devuelve el elemento más antiguo de la cola C y no
  modifica la cola }

principio
    devuelve( C.datos[C.primer0] )
fin

accion eliminar( e/s Cola C )
{ Modifica la cola C eliminando el elemento más antiguo }

principio
    C.num ← C.num - 1
    C.primer0 ← suma_uno(C.primer0)
fin

funcion suma_uno( entero n ) devuelve entero
{ Suma uno a la posición n en el sentido circular de las
  agujas del reloj }

principio
    devuelve( (n+1) MOD MAXCOLA )
fin
```

Nota: esta implementación es válida para colas de como mucho MAXCOLA elementos

2b.- IMPLEMENTACIÓN DINÁMICA:

• **REPRESENTACIÓN:**

```
tipo
    Nodo = registro
        telemento dato
        puntero a Nodo sig
    freg;

Cola = registro
    puntero a Nodo primero
    puntero a Nodo ultimo
freg;
```

• **INTERPRETACIÓN:**

Cola es un registro con dos punteros de forma que el puntero `primero` apunta a un nodo que contiene el elemento más antiguo de la cola y un puntero que apunta al nodo que contiene el elemento que llegó a la cola después. El puntero del nodo correspondiente al elemento más reciente de la cola apunta a `NULL`. El puntero `ultimo` apunta al nodo en el que está el elemento más reciente de la cola.

• **IMPLEMENTACIÓN DE LAS OPERACIONES:**

```
accion iniciarCola( sal Cola C )
{ Inicia C como una cola vacía }

principio
    C.primero ← NULL
    C.ultimo ← NULL
fin

accion añadir( e/s Cola C, ent telemento d )
{ Añade a C el elemento d }

variables
    nuevo : puntero a Nodo
principio
    nuevo ← reservar(Nodo)
    si nuevo ≠ NULL entonces
        dest(nuevo).dato ← d
        dest(nuevo).sig ← NULL
        si C.primero = NULL entonces
            C.primero ← nuevo
        sino
            dest(C.ultimo).sig ← nuevo
        fsi
    C.ultimo ← nuevo
    fsi
fin
```

```
funcion colaVacía( Cola C ) devuelve booleano  
{Si C está vacía devuelve VERDAD y FALSO en otro caso}
```

```
principio  
    devuelve( C.primerO = NULL )  
fin
```

```
funcion primero( Cola C ) devuelve telemento  
{ Devuelve el elemento más antiguo de la cola C y no  
  modifica la cola }
```

```
principio  
    devuelve( dest(C.primerO).dato )  
fin
```

```
accion eliminar( e/s Cola C)  
{ Modifica la cola C eliminando el elemento más antiguo }
```

```
variables  
    aux : puntero a Nodo  
principio  
    aux ← C.primerO  
    C.primerO ← dest(C.primerO).sig  
    liberar(aux)  
    si C.primerO = NULL entonces  
        C.ultimo ← NULL  
    fsi  
  
fin
```