

ESTRUCTURA DE DATOS COLA

UNA COLA ES UN **CONJUNTO ORDENADO** DE ELEMENTOS **HOMOGÉNEOS**, EN EL CUAL LOS ELEMENTOS SE ELIMINAN POR UNO DE LOS EXTREMOS, DENOMINADOS **CABEZA**, Y SE AÑADEN POR EL OTRO EXTREMO, DENOMINADO **FINAL**. LAS ELIMINACIONES Y AÑADIDOS SE REALIZAN SIGUIENDO UNA **POLÍTICA FIFO** (FIRST IN, FIRST OUT).

CONJUNTO ORDENADO: LOS ELEMENTOS DE LA COLA SE SITÚAN SIGUIENDO UN CIERTO ORDEN, NO ESTÁN ORDENADOS EN FUNCIÓN DE SU VALOR.

HOMOGENEA: TODOS LOS ELEMENTOS SON DEL MISMO TIPO.

DINÁMICA: SU TAMAÑO VARÍA CUANDO SE REALIZA UNA OPERACIÓN DE APILAR O DESAPILAR.

POLITICA FIFO: SE DENOMINA AL MODO DE FUNCIONAMIENTO EN EL CUAL EL PRIMER ELEMENTO QUE SE AÑADE A LA ESTRUCTURA ES EL PRIMERO EN SALIR.

CABEZA: ES EL PRINCIPIO O FRENTE DE LA MISMA.

FINAL: ES EL FONDO DE LA COLA.

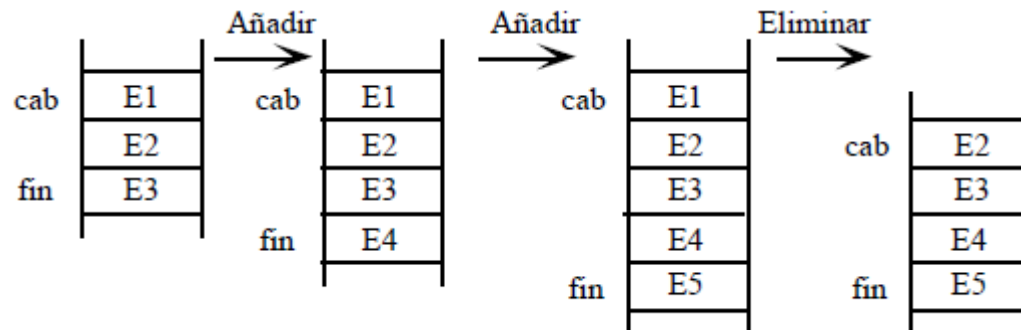
ESTRUCTURA DE DATOS COLA

UNA COLA ES UN **CONJUNTO ORDENADO** DE ELEMENTOS **HOMOGÉNEOS**, EN EL CUAL LOS ELEMENTOS SE ELIMINAN POR UNO DE LOS EXTREMOS, DENOMINADOS **CABEZA**, Y SE AÑADEN POR EL OTRO EXTREMO, DENOMINADO **FINAL**. LAS ELIMINACIONES Y AÑADIDOS SE REALIZAN SIGUIENDO UNA **POLÍTICA FIFO** (FIRST IN, FIRST OUT).

SOBRE UNA COLA PODEMOS REALIZAR DOS OPERACIONES BÁSICAS DE MANIPULACIÓN:

AÑADIR: AÑADE UN ELEMENTO AL FINAL DE LA COLA.

ELIMINAR: ELIMINA UN ELEMENTO DE LA CABEZA DE LA COLA.



ESTRUCTURA DE DATOS COLA

IMPLEMENTACIÓN ESTÁTICA

IMPLEMENTACIÓN MEDIANTE VECTORES:

Para realizar esta implementación, se toma como modelo la implementación del tipo Cola mediante vectores.

Pero es conveniente utilizar una sola entidad para contener la estructura de datos, y por tanto utilizar un registro e integrar el vector como uno de sus campos.

Además del vector, se necesitan unos índices que indiquen en cada momento qué elemento de dicho vector se está comportando como elemento Cabeza y qué elemento se comporta como elemento Final.

Por lo tanto, el registro estará constituido por tres campos: un vector conteniendo los elementos de la cola, y dos valores, cabeza y final, que definen las posiciones del vector asociadas a ambos elementos de la cola.

ESTRUCTURA DE DATOS COLA

LA ESTRUCTURA DE DATOS PARA IMPLEMENTAR UNA COLA, SIGUIENDO EL ESQUEMA ANTERIOR ES:

MaxElemen = ... {nº máximo de elementos en la cola}

TipoCola = REGISTRO

datos: ARREGLO [0. . MaxElemen] de TipoDato

cabeza, final: -1. . MaxElemen

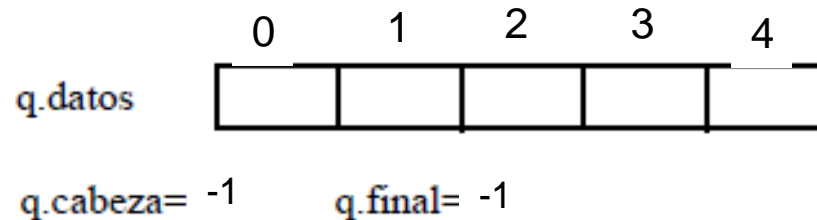
FINREGISTRO

Q: TipoCola

ESTRUCTURA DE DATOS COLA

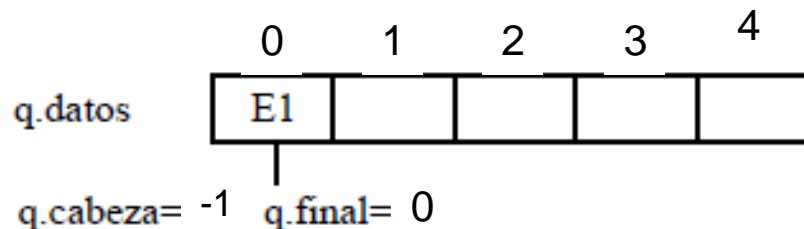
Antes de ver la implementación concreta de las operaciones, es conveniente realizar una pequeña discusión sobre su funcionamiento. Se parte, como en el caso de la Pila, de que al realizar la operación **CrearCola**, tanto cabeza como final indicarán la posición -1:

CREARCOLA(Q):



AÑADIR(Q, E1):

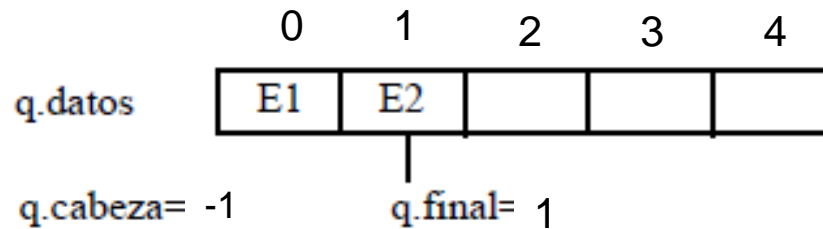
Como final indica la última posición ocupada, una política sencilla es añadir en la siguiente ($\text{final}+1$); por lo tanto, la operación Añadir supondrá incrementar final y añadir el elemento en la posición indicada: $Q.\text{datos}[q.\text{final}]$



ESTRUCTURA DE DATOS COLA

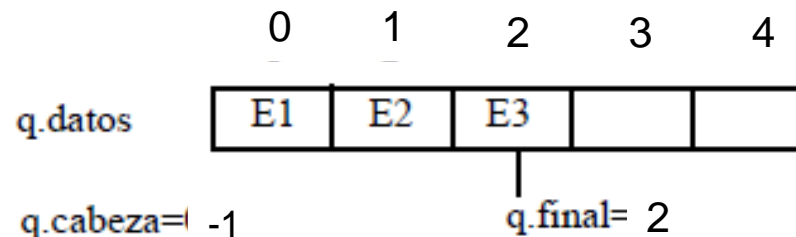
AÑADIR(Q, E2):

SIGUIENDO LA DEFINICIÓN ANTERIOR DE LA OPERACIÓN:



AÑADIR(Q, E3):

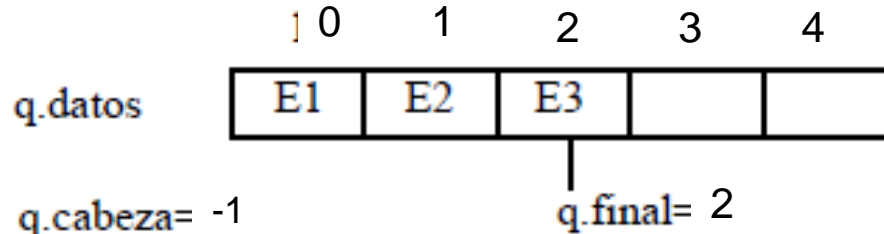
SIGUIENDO LA DEFINICIÓN ANTERIOR DE LA OPERACIÓN:



ESTRUCTURA DE DATOS COLA

CABEZA(Q):

El elemento más antiguo en la cola es **E1**, que ocupa la posición 0. El valor de cabeza es -1, luego esta operación devolverá el elemento situado en la siguiente posición a la indicada por la cabeza: $q.datos[q.cabeza+1]$, que en este caso es **E1**.

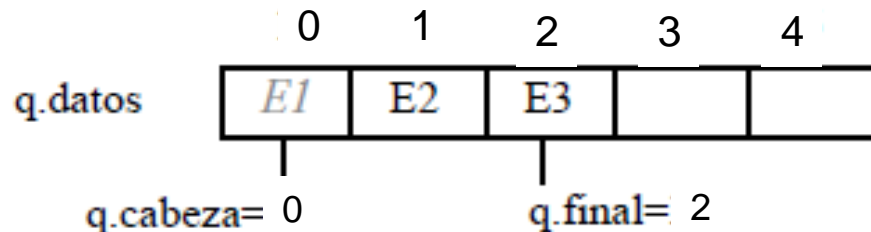


Esta operación no modifica ninguno de los componentes de la cola; ni siquiera el valor de los campos cabeza y final.

ESTRUCTURA DE DATOS COLA

ELIMINAR(Q):

Dado que, como hemos dicho anteriormente, el campo cabeza indica la posición anterior al elemento más antiguo de la cola, la operación Eliminar supondrá eliminar de la cola el elemento situado en la posición $q.cabeza+1$. Para ello bastará con incrementar el valor del campo cabeza para que "apunte" al siguiente elemento de la cola. El elemento eliminado en este caso es el E1, y el más antiguo en la cola pasa a ser E2.

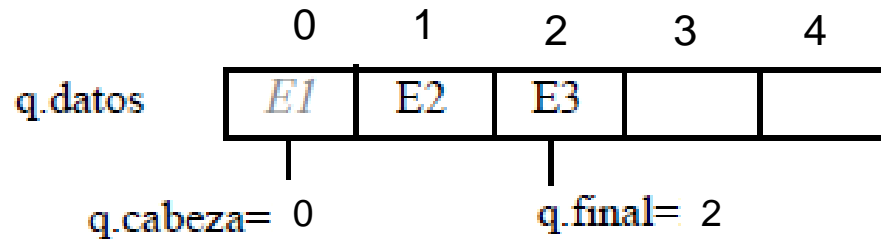


No se ha sobrescrito el valor de E1 y este sigue en la posición 0 del vector. Sin embargo, desde un punto de vista lógico, SE HA eliminado este elemento de la cola. En este momento no existe ninguna operación válida que permita acceder al mismo. En la figura los elementos eliminados están en cursiva. Estos elementos ya no forman parte de la cola.

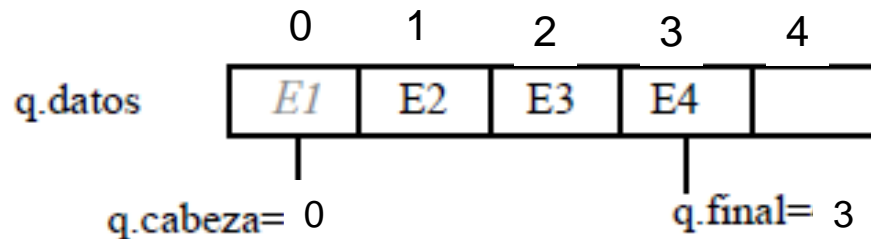
ESTRUCTURA DE DATOS COLA

CABEZA(Q):

AL DEVOLVER $Q.DATOS[Q.CABEZA+1]$, EL RESULTADO ES E2.

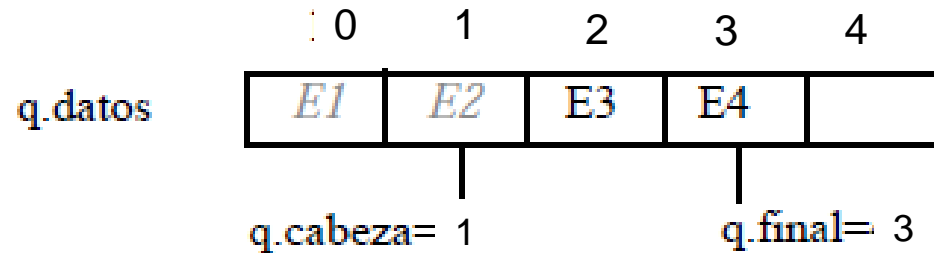


AÑADIR(Q, E4):

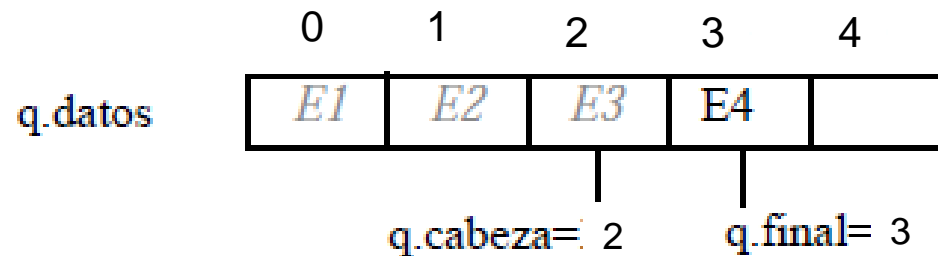


ESTRUCTURA DE DATOS COLA

ELIMINAR(Q):

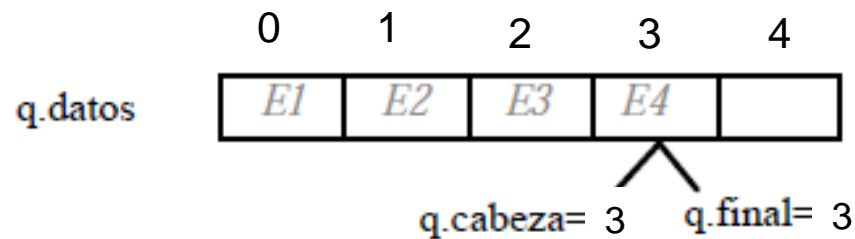


ELIMINAR(Q):

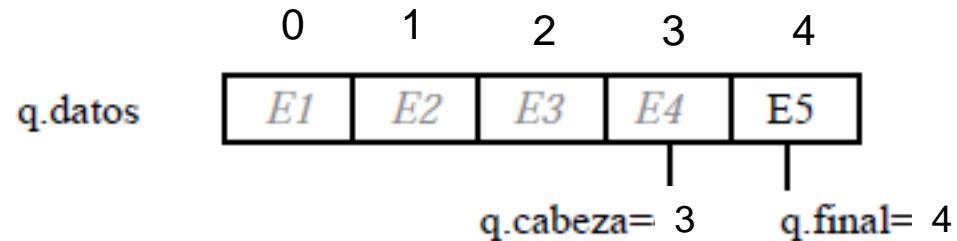


ESTRUCTURA DE DATOS COLA

ELIMINAR(Q):

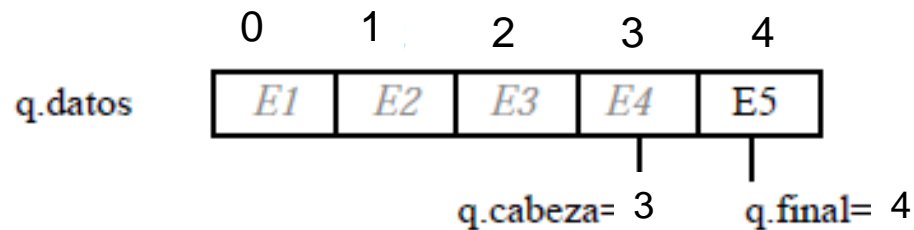


AÑADIR(Q, E5):



ESTRUCTURA DE DATOS COLA

Pero, ¿cuál es la condición de Cola Llena? Podemos utilizar la condición ($q.final=MAX$) para afirmar que la cola está llena, tal y como ocurre en la siguiente figura:



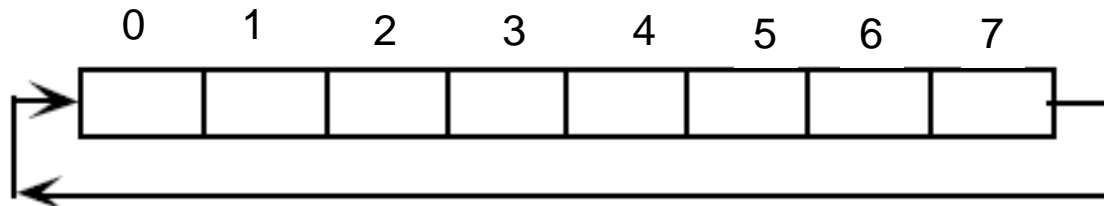
Sin embargo, si utilizamos la condición anterior, se da la paradoja de que hay sitio en el vector, pero no se puede añadir ningún elemento más.

¿CUÁL ES LA SOLUCIÓN?

ESTRUCTURA DE DATOS COLA

La solución implica reutilizar las componentes del vector que contenían elementos ya eliminados. Esto es, cuando durante el proceso de añadir se llegue al final del vector, se comienza a llenar de nuevo las componentes iniciales del mismo, si se encuentran vacías.

Para lograr esto, se maneja el vector como si fuese un "vector circular". Esto significa que no se considera el componente **MAX** del vector como el último del mismo, sino que se considera que el siguiente componente del último elemento, es otra vez el primer elemento del vector. Gráficamente:



ESTRUCTURA DE DATOS COLA

La definición de la Estructura Cola circular sería la siguiente:

Max = ... {n° máximo de elementos en la cola}

TipoCola = REGISTRO

 datos: ARREGLO [0. . Max+1] de TipoDato

 cabeza, final: -1. . Max

FINREGISTRO

Q: TipoCola

Al utilizar un vector circular para la implementación, la siguiente componente a la posición POS en el vector no siempre será $i+1$, sino que vendrá dada por la función:

FUNCIÓN SIGUIENTE (POS:ENTERO):ENTERO;

COMIENZO

SI (POS <> Max)

ENTONCES

Siguiente := POS+1

SINO

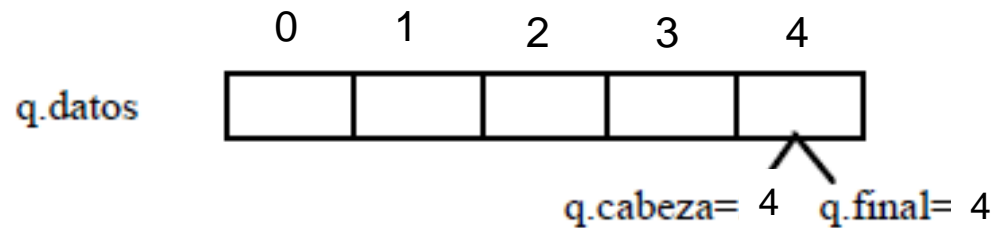
Siguiente := 0

FINSI

FIN

ESTRUCTURA DE DATOS COLA

```
PROCESO CrearCola ( VAR q: TipoCola );  
COMIENZO  
q.cabeza := Max;  
q.final := Max  
FIN;
```



ESTRUCTURA DE DATOS COLA

En un vector circular, la posición anterior a la primera en el vector es la última, por lo que q.cabeza se inicializa a MAX. Por otro lado, consideramos que la cola se encuentra vacía cuando coinciden q.cabeza y q.final. Así pues, también debemos inicializar q.final en MAX.

FUNCIÓN ColaVacía (q: TipoCola): BOOLEAN;

SI q.cabeza = q.final

ENTONCES

ColaVacía = true

SINO

ColaVacía = false

FINSI

FIN;

La cola se considera vacía cuando la cabeza y el final de la misma apuntan a la misma posición del vector circular. Esta condición también se da con una cola recién creada.

ESTRUCTURA DE DATOS COLA

El elemento cabeza de la cola es el que ocupa la siguiente posición a la apuntada por el campo q.cabeza. Además esta operación devuelve un error cuando la cola está vacía.

PROCESO Cabeza (q: TipoCola ; VAR e:Tipodato);

COMIENZO

e := q.datos[Siguiente(q.cabeza)]

FIN;

ESTRUCTURA DE DATOS COLA

Para eliminar un elemento de la cola basta con hacer que el campo q.cabeza apunte al siguiente elemento del vector.

PROCESO Eliminar (VAR q: TipoCola)

e = q.dato (Siguiente(q.cabeza))

q.cabeza := Siguiente(q.cabeza)

FIN;

ESTRUCTURA DE DATOS COLA

Para añadir un elemento, se hace que el campo q.final apunte al siguiente de la cola y se coloca en esa posición. Obviamente no podremos añadir un elemento a la cola si está llena.

PROCESO Añadir (VAR q: TipoCola; e:Tipodato);

COMIENZO

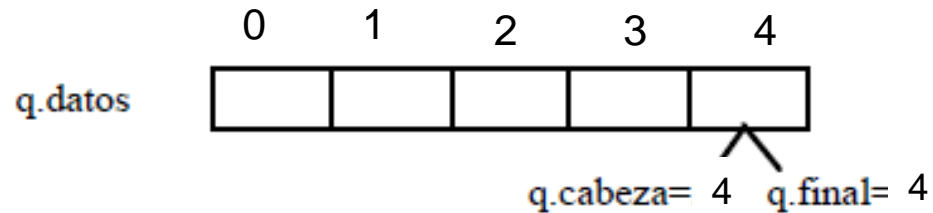
q.final := Siguiente(q.final);

q.datos[q.final] := e

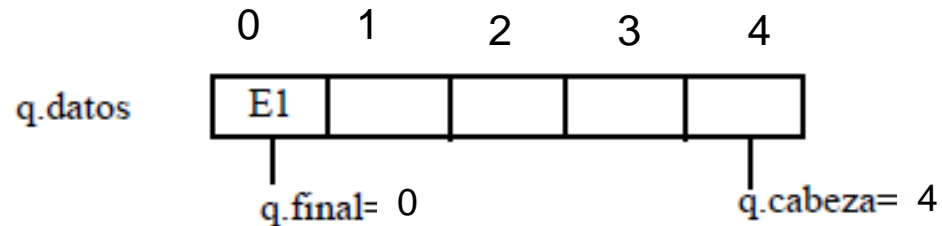
FIN;

ESTRUCTURA DE DATOS COLA

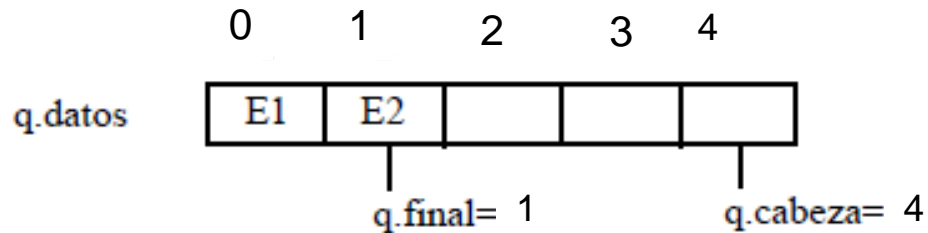
CREARCOLA(Q):



AÑADIR(Q, E1): SIGUIENTE(Q.FINAL) = 0

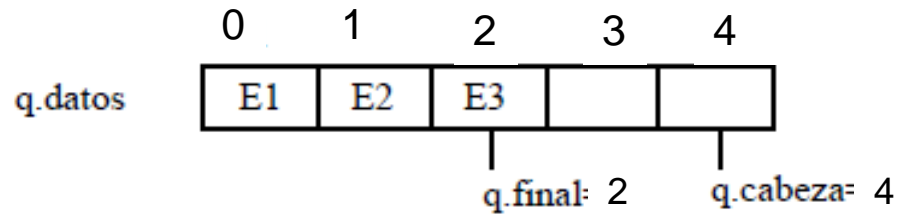


AÑADIR(Q, E2): SIGUIENTE(Q.FINAL) = 1

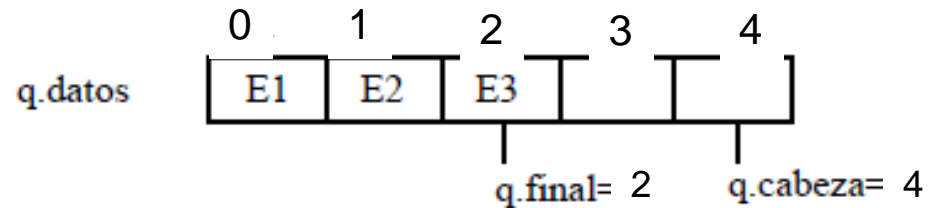


ESTRUCTURA DE DATOS COLA

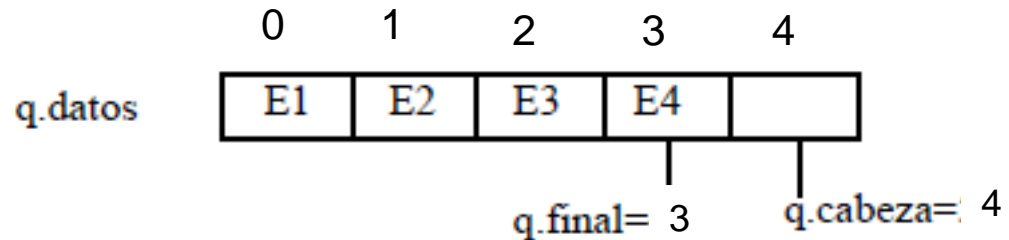
AÑADIR(Q, E3):SIGUIENTE(Q.FINAL) = 2



CABEZA(Q): SIGUIENTE(Q.CABEZA) = 0; CABEZA VALE E1



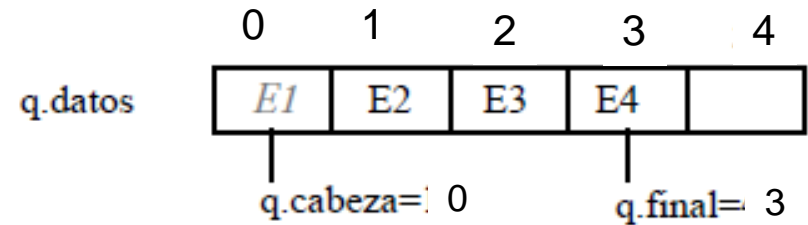
AÑADIR(Q, E4): SIGUIENTE(Q.FINAL) = 3



ESTRUCTURA DE DATOS COLA

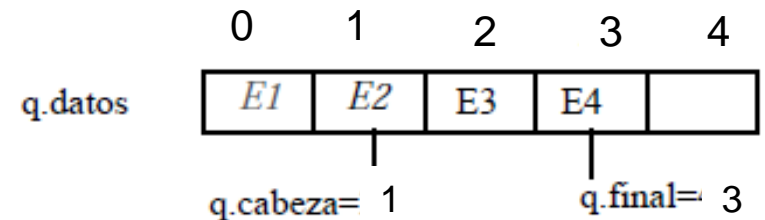
ELIMINAR(Q): SIGUIENTE(Q.CABEZA) = 0

E = E1

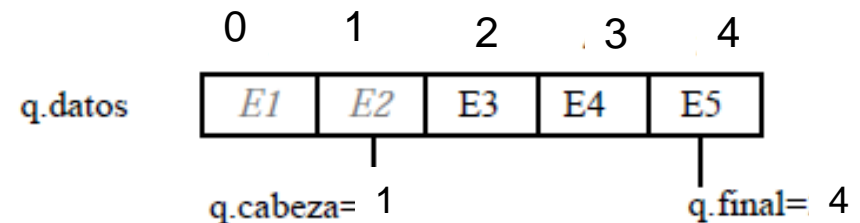


ELIMINAR(Q): SIGUIENTE(Q.CABEZA) = 1

E = E2

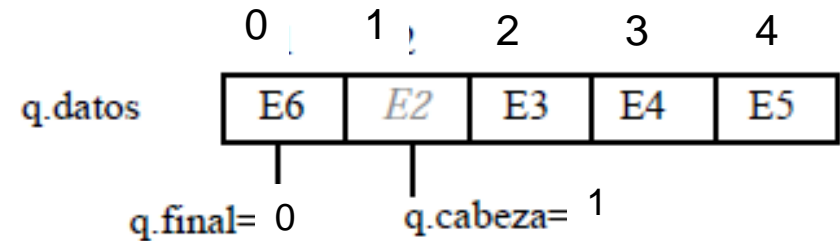


AÑADIR(Q, E5): SIGUIENTE(Q.FINAL) = 4

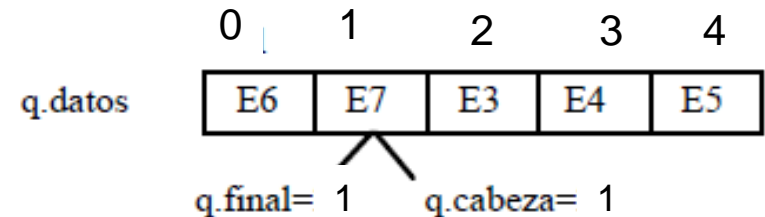


ESTRUCTURA DE DATOS COLA

AÑADIR(Q, E6): SIGUIENTE(Q.FINAL) = 0



AÑADIR(Q, E7): SIGUIENTE(Q.FINAL) = 1



En este punto, ¿la cola está Llena o Vacía?. Sabemos que se cumple la condición de cola vacía porque q.final y q.cabeza son iguales. Sin embargo, tal y como se ha desarrollado el ejemplo, sabemos que la cola está llena. ENTONCES hay que establecer algún mecanismo para diferenciar ambas situaciones.

ESTRUCTURA DE DATOS COLA

La solución consiste en dejar siempre una posición libre en la Cola. Esta posición se encontrará en la componente apuntada por la cabeza.

Esta componente es justo la anterior en el vector circular a la ocupada por el elemento más antiguo de la cola. En esta implementación, esta posición será especial, dado que no podrá ser ocupada por ningún elemento.

Con esta restricción, la condición de cola llena se producirá cuando el final de la cola se encuentre justo antes de esta posición (q.cabeza), es decir, cuando el siguiente elemento debería añadirse en la misma. Por lo tanto, la definición de Cola Llena queda de la siguiente forma:

```
FUNCIÓN ColaLlena ( q: TipoCola ): BOOLEAN;  
COMIENZO  
SI (q.cabeza = Siguiente(q.final))  
    ENTONCES  
        ColaLlena =true  
SINO  
        ColaLlena = false  
FINSI  
FIN
```


ESTRUCTURA DE DATOS COLA

Con esta definición de cola llena, la última operación del ejemplo anterior habría devuelto un error. La condición de cola llena se habría cumplido y no se habría podido añadir el elemento E7.

Obviamente, la solución que hemos utilizado con un vector circular desaprovecha una de las posiciones del vector, pero es mejor que la que usa un vector lineal que acaba llenándose con mucha mayor facilidad y desperdiciaba más posiciones.

ESTRUCTURA DE DATOS COLA

EJERCICIOS:

- 1- Crear una cola T con capacidad de 20 elementos y cargarla con números reales, luego mostrar los elementos de la cola. Mantener la Cola original**
- 2- Dada una Cola C de N elementos, leer un número, si existe eliminarlo de la cola, caso contrario mostrar un mensaje «Número Inexistente».**