2.2 Colas, explicación y ejemplos.

Los humanos para realizar una compra, un trámite, o algo en una oficina, tenemos la costumbre de formarnos para esperar turno haciendo cola, en una cola siempre habrá un primero y un último en la misma. En la memoria RAM la idea es simular o emular con datos una cola formada por los humanos.

Se usará un arreglo de cierto tipo para simular una cola, por ejemplo, haremos una cola de números reales, (dobles en Java), la idea es formar los números reales uno tras otro. Se usarán dos apuntadores: primero y último para señalar el primero y el último elemento presentes en la cola, la localidad cero del arreglo no se usa para evitar ambigüedades.

Las declaraciones globales para implementar una cola simple son:

Entero (int) tamaño=30; real (doublé) base [tamaño];

Entero (int) primero=0; entero (int) ultimo=0; indicando que la cola está vacía

Se tiene:

1 2 3 4 5 6 7 8 … _tamaño

_primero=0 _último=0

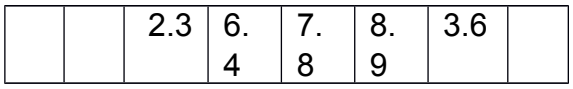
Ejemplos

1. Supongamos que tenemos la cola de caracteres que se muestra en el arreglo:

1 2 3 4 5 6 7 … tamaño

Donde se tiene: primero=1 y último=6

2. Supongamos que tenemos la cola de números enteros declarada como: entero base [8], el estado actual de esa cola se muestra en el arreglo:



1 2 3 4 5 6 7 8=tamaño

Se tiene así: primero=3 y último=7

Una cola se declara en la forma: tipo base [tamaño]; donde tipo es un tipo de datos válido, base es el nombre del arreglo, y tamaño es el tamaño del arreglo, que un número entero.

Los algoritmos para agregar datos a una cola, y retirar datos de una cola se declaran en la forma que sigue:

Agregar (tipo dato)

```
If ultimo=tamaño
    Escribe cola llena
else  if primero=0 y ultimo=0 // cola vacía
         _primero=1, ultimo=1, base [ultimo]=dato
         else ultimo=ultimo+1  // cola no está vacía ni llena
            base [ultimo]=dato
fin
```

Un algoritmo para poder ver en cualquier momento el estado actual de la cola es:

```
estado Actual ()
if primero=0 y ultimo=0
       Escribe “cola Vacía” // cola vacía
else  entero k=primero
      while (k<=ultimo)
          _escribe  base[k],
          k=k+1
    fin_ while
fin
```

Un algoritmo para retirar de la cola es:

```
_tipo dato retirar ( )
If primero=0 y ultimo=0
      Escribe “cola Vacía”
else  if primero=ultimo // solo hay un elemento en la cola
            _dato=base [primero],
            _ultimo=0, primero=0,
            _regresa dato
   else  dato=base [primero], primero=primero+1,
            _regresa dato
fin
```

Observación: Al operar una cola del tipo que sea, hay un desplazamiento a la derecha de sus elementos sobre el arreglo base, es decir, se genera un desplazamiento a la derecha de los elementos sobre el arreglo, generando una zona vacía en la izquierda del arreglo base.

Hay varias formas de compensar eso, en la teoría clásica de colas se permite que la rutina agregar y la rutina retirar, trabajen sobre la parte del arreglo que va quedando vacía, generando así la llamada cola circular.

W	W	D	t	z	I	J	Y	R	E
1	2	3	4	5	6	7	8		

tamaño

_primero=5, ultimo=4

Una vez que el último alcanza tamaño no es posible agregar en una cola simple, se genera la cola circular permitiendo que el último de vuelta al arreglo, y agregue cosas por las localidades que van quedando vacías en la parte izquierda del arreglo.

Condiciones en que la cola circular queda llena:

a): primero=1 y ultimo=tamaño-1;

b): ultimo=primero-1, o bien, primero=ultimo+1

Un algoritmo para agregar cosas en una cola circular es

_entero tamaño; tipo base [tamaño]; entero primero=0; entero ultimo=0;

Agregar (tipo dato)

Si (primero=1 y ultimo=tamaño) o (primero=ultimo+1)

Escribe cola circular llena

else Si ultimo=0 y primero =0 // cola vacía

{_primero=1, ultimo=1, base [ultimo]=dato;}

else si (ultimo=tamaño) {ultimo=1, base[ultimo]=dato;}

else {ultimo=ultimo+1, base[ultimo]=dato;}

_fin.

Un algoritmo para ver el estado actual de la cola circular es:

Estado Actual ()

Si (primero=0 y ultimo=0) escribe cola circular vacía

else k=primero

```
    while( k != ultimo)
    {_escribe base[k];
       if (k=tamaño) k=1 else    k=k+1 }
fin.
```

Un algoritmo para retirar de la cola circular es:

```
Tipo retirar ()
Si (primero=0 y ultimo=0)  escribe cola circular vacía
else si (primero=ultimo)
          {_dato=base [primero]; primero=0; ultimo=0; regresa dato}
      else
           _si (primero=tam-1) {dato=base [primero]; primero=1; regresa dato}
     else {dato=base[primero]; primero=primero+1; regresa dato;}
_fin
```