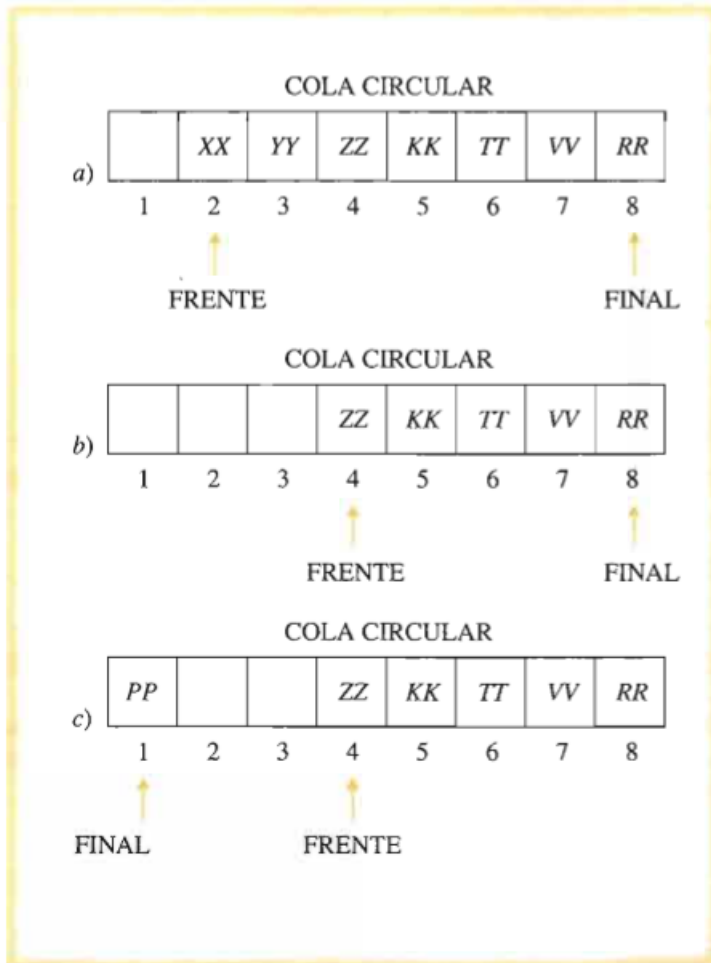


Colas circulares

Una cola circular constituye una estructura de datos lineal en la cual el siguiente elemento del último en realidad es el primero. De esta forma se utiliza de manera más eficiente la memoria de la computadora.

-Cola con un array circular

La alternativa, sugerida en la operación quitar un elemento, de desplazar los restantes

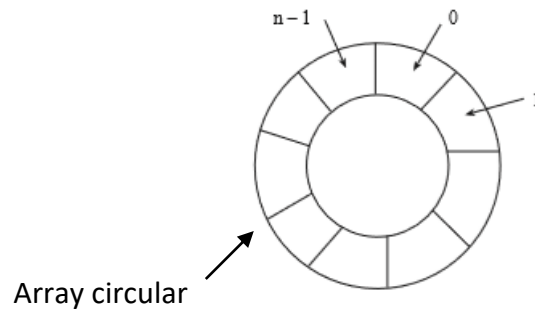


Cola con un array circular

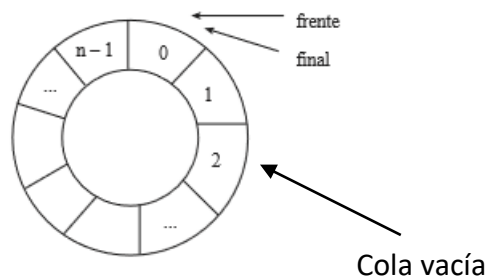
La alternativa, sugerida en la operación *quitar* un elemento, de desplazar los restantes elementos del array de modo que la *cabeza* de la cola vuelva al principio del array, es costosa, en términos de tiempo de computadora, especialmente si los datos almacenados en el array son estructuras de datos grandes.

La forma más eficiente de almacenar una cola en un array es modelar éste de tal forma que se una el extremo final con el extremo cabeza. Tal array se denomina *array circular* y

permite que la totalidad de sus posiciones se utilicen para almacenar de la cola sin necesidad de desplazar elementos.



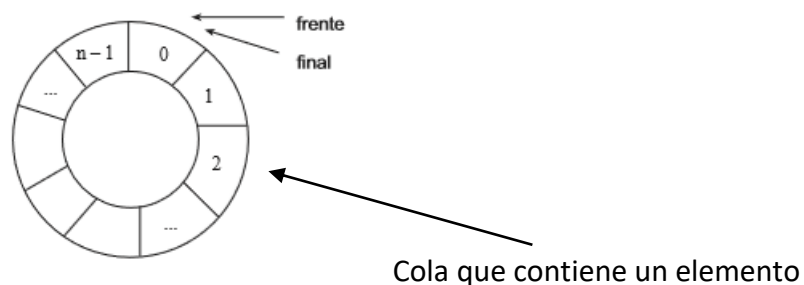
El array se almacena de modo natural en la memoria, como un bloque lineal de n elementos. Se necesitan dos marcadores (apuntadores) *frente* y *final* para indicar, respectivamente, la posición del elemento *cabeza* y la posición donde se almacenó el último elemento puesto en la cola.



El apuntador *frente* siempre contiene la posición del primer elemento de la cola y avanza en el sentido de las agujas del reloj; *final* contiene la posición donde se puso el último elemento, también avanza en el sentido del reloj (circularmente a la derecha). La implementación del movimiento circular se realiza según la teoría de los restos, de tal forma que se generen índices de 0 a $\text{MAXTAMQ} - 1$:

Mover *final* adelante = $(\text{final} + 1) \% \text{MAXTAMQ}$

Mover *frente* adelante = $(\text{frente} + 1) \% \text{MAXTAMQ}$



La implementación de la gestión de colas con un array circular ha de incluir las operaciones básicas del TAD Cola, es decir, las siguientes tareas básicas:

° Creación de una cola vacía, de tal forma que final apunte a una posición inmediatamente anterior a frente:

frente = 0; final = MAXTAMQ – 1

° Comprobar si una cola está vacía:

frente == siguiente (final)

° Comprobar si una cola está llena. Para diferenciar la condición cola llena de cola vacía se sacrifica una posición del array, entonces la capacidad real de la cola será ser MAXTAMQ-1. La condición de cola llena:

frente == siguiente (siguiente(final))

° Poner un elemento a la cola: si la cola no está llena, fijar final a la siguiente posición:

final = (final + 1) % MAXTAMQ y asignar el elemento.

° Retirar un elemento de la cola: si la cola no está vacía, quitarlo de la posición frente y establecer frente a la siguiente posición: (frente + 1) % MAXTAMQ

° Obtener el elemento primero de la cola, si la cola no está vacía, sin suprimirlo de la cola.

Referencias:

- Cairó, O., Guardati, S., (2006). *Estructura de Datos*. McGraw Hill

- Joyanes, L., Sánchez, L., Zahonero, I., (2007). *Estructura de datos en C++*. McGraw Hill