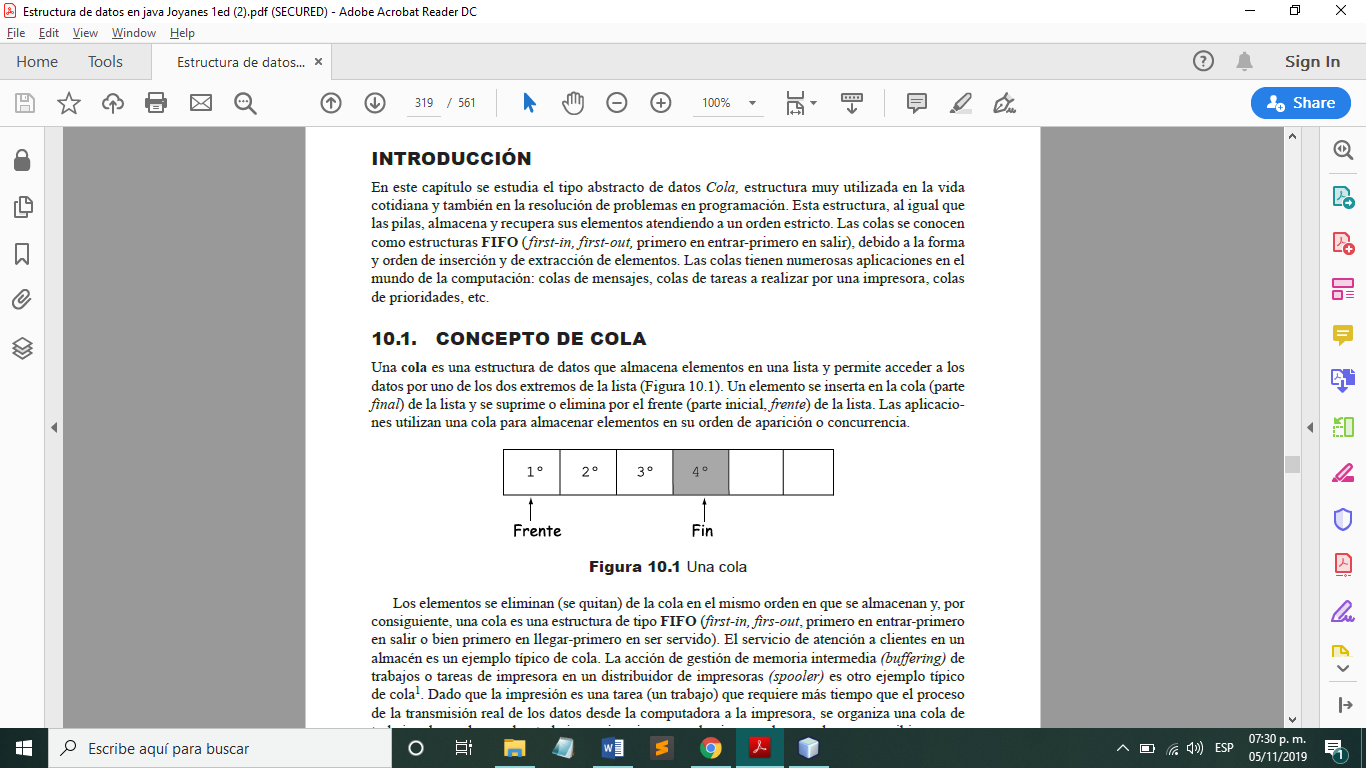
**COLAS CIRCULARES**

**CONCEPTO**

**¿Qué son las colas?**

Una cola es una estructura de datos cuyos elementos mantienen un cierto orden, de tal modo que sólo se pueden añadir elementos por un extremo, **final** de la cola, y eliminar o extraer por el otro extremo, llamado **frente**.

Una cola es una estructura de datos que almacena elementos en una lista y permite acceder a los datos por uno de los dos extremos de la lista.



Un elemento se inserta en la cola (parte final) de la lista y se suprime o elimina por el frente (parte inicial, frente) de la lista. Las aplicaciones utilizan una cola para almacenar elementos en su orden de aparición o concurrencia.

**APLICACIONES EN LA VIDA REAL DE LAS COLAS**

**¿Para qué sirven las colas?**

Las colas se conocen como estructuras FIFO (first-in, first-out, primero en entrar-primero en salir), debido a la forma y orden de inserción y de extracción de elementos. Las colas tienen numerosas aplicaciones en el mundo de la computación: colas de mensajes, colas de tareas a realizar por una impresora, colas de prioridades, etc.

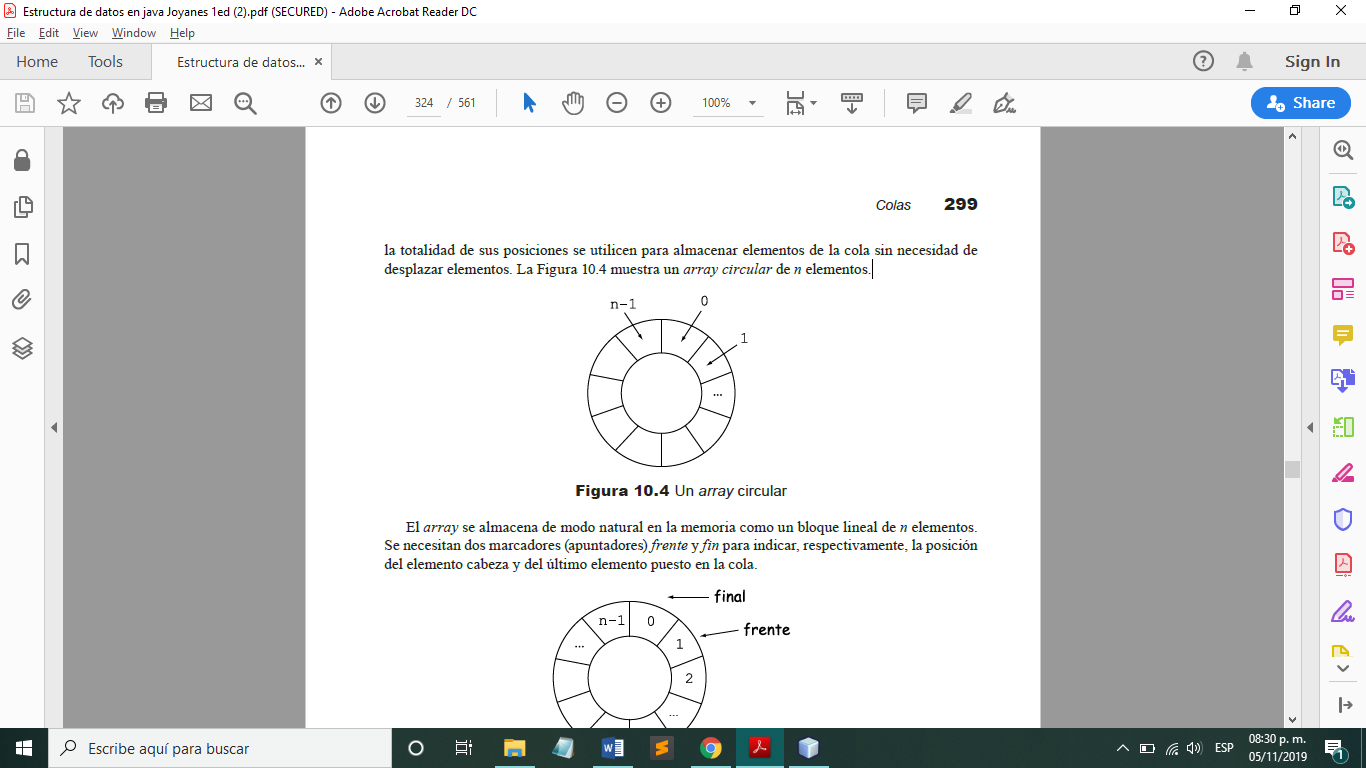
El servicio de atención a clientes en un almacén es un ejemplo típico de cola. La acción de gestión de memoria intermedia *(buffering)* de trabajos o tareas de impresora en un distribuidor de impresoras *(spooler)* es otro ejemplo típico de cola. Dado que la impresión es una tarea (un trabajo) que requiere más tiempo que el proceso de la transmisión real de los datos desde la computadora a la impresora, se organiza una cola de trabajos de modo que los trabajos se imprimen en el mismo orden en el que se recibieron por la impresora. Este sistema tiene el gran inconveniente de que si su trabajo personal consta de una única página para imprimir y delante de su petición de impresión existe otra petición para imprimir un informe de 300 páginas, deberá esperar a la impresión de esas 300 páginas antes de que se imprima su página.

**OPERACIONES QUE SIRVEN PARA DEFINIR UNA COLA**

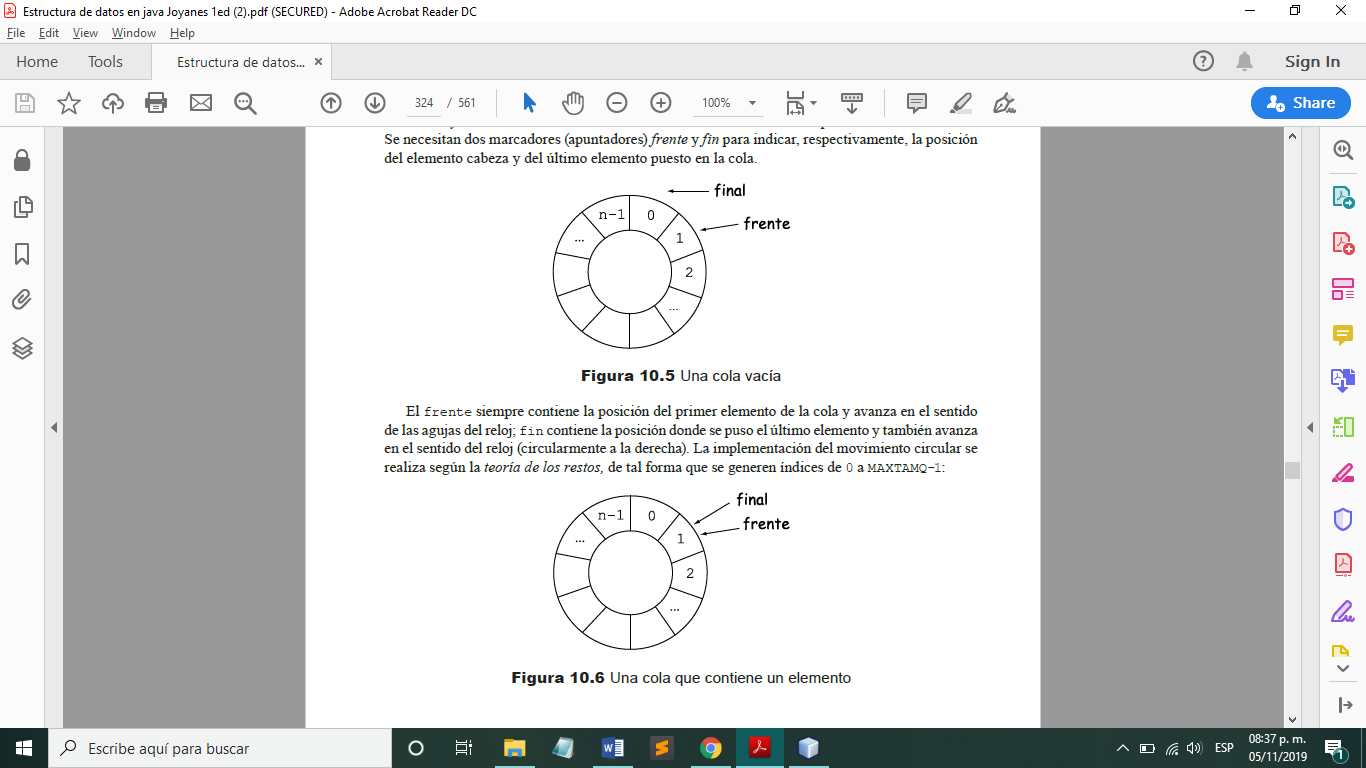
Las operaciones que sirven para definir una cola y poder manipular su contenido son las siguientes:

|  |  |
| --- | --- |
| Tipo de dato | Elemento que se almacena en la cola |
| Operaciones |  |
| *Crear Cola* | Inicia la cola como vacía |
| *Insertar* | Añade un elemento por el final de la cola |
| *Quitar* | Retira (extrae) el elemento frente de la cola |
| *Cola vacía* | Comprueba si la cola está llena de elementos |
| *Frente* | Obtiene el elemento frente o primero de la cola |
| *Tamaño de la cola* | Número de elementos máximo que puede contener la cola |

La forma más eficiente de almacenar una cola en un *array* es modelarlo 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 elementos de la cola sin necesidad de desplazar elementos. La Figura 10.4 muestra un *array circular* de *n* elementos.



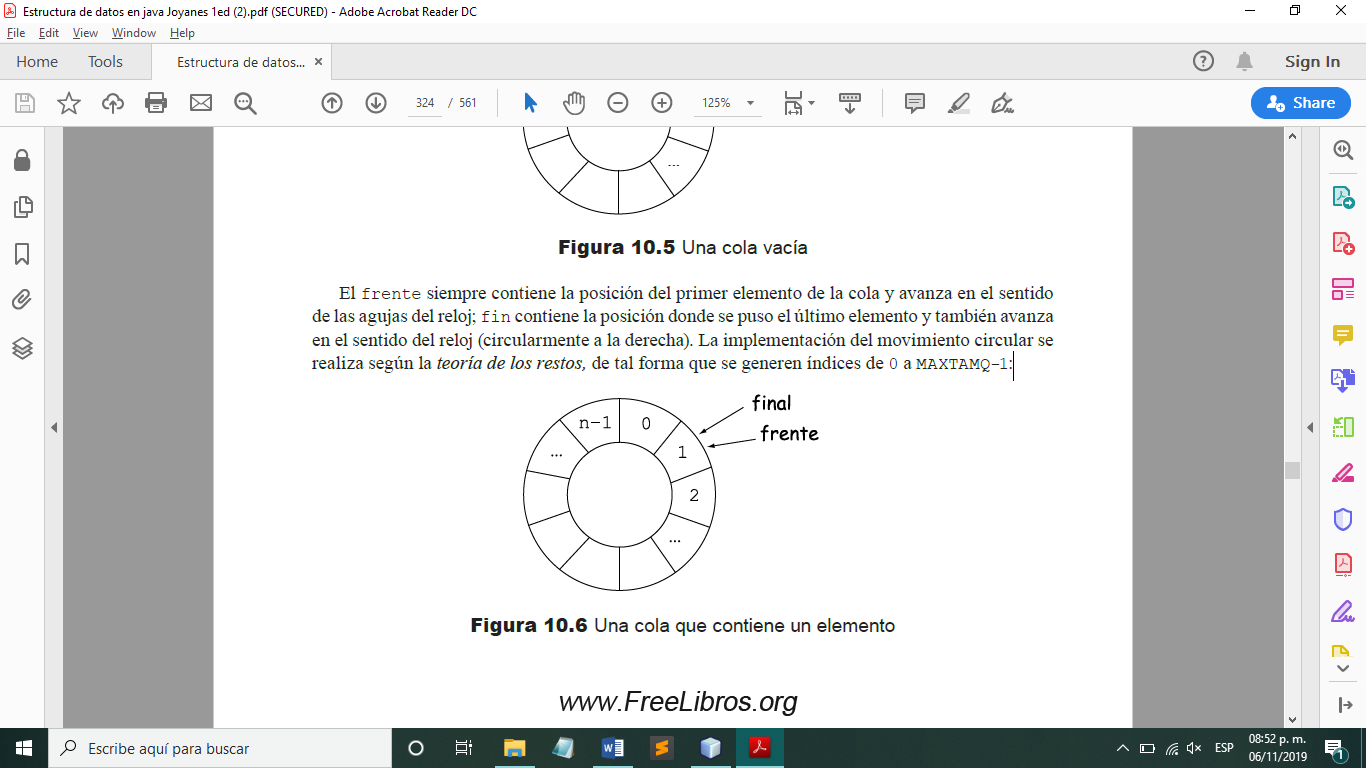
El *array* se almacena de modo natural en la memoria como un bloque lineal de *n* elementos. Se necesitan dos marcadores (apuntadores) *frente* y *fin* para indicar, respectivamente, la posición del elemento cabeza y del último elemento puesto en la cola.



**Las colas circulares permiten la reutilización de espacio que se desperdicia en una cola no circular.**

**Una cola circular o anillo es una estructura de datos en la que los elementos están de forma circular y cada elemento tiene un sucesor y un predecesor. Los elementos pueden consultarse, añadirse y eliminarse únicamente desde la cabeza del anillo que es una posición distinguida.**

El frente siempre contiene la posición del primer elemento de la cola y avanza en el sentido de las agujas del reloj; **fin** contiene la posición donde se puso el último elemento y 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 MAXTAMQ-1:



Los algoritmos que formalizan la gestión de colas en un *array* circular han de incluir las operaciones básicas del *TAD Cola*, en concreto, las siguientes tareas básicas:

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

*frente = 0; fin = MAXTAMQ-1.*

* Comprobar si una cola está vacía:

*frente == siguiente (fin)*

* Comprobar si una cola está llena. Para diferenciar la condición de *cola llena* de *cola vacía* se sacrifica una posición del *array,* de tal forma que la capacidad de la cola va a ser *MAXTAMQ-1*. La condición de cola llena es:

*frente == siguiente (siguiente (fin))*.

* Poner un elemento a la cola: si la cola no está llena, avanzar fin a la siguiente posición, *fin = (fin + 1) % MAXTAMQ*, y asignar el elemento.
* Retirar un elemento de la cola: si la cola no está vacía, quitarlo de la posición *frente* y avanzar *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.

RECUPERADO DE: Estructura de datos en Java por Joyanes, 1era Edición.