

TD COLA (QUEUE)

Una **cola** es un contenedor de objetos que se insertan y se eliminan siguiendo el principio 'Primero en entrar, primero en salir' (F.I.F.O.= 'First In, First Out').

En ese sentido puede decirse que una cola es una lista con restricciones en el alta (encolar o enqueue) y baja (desencolar, dequeue).

El Frente (FRONT) de la cola corresponde al elemento que está en primer lugar, es decir que es el que estuvo más tiempo en espera.

El Fondo (END) de la cola corresponde al último elemento ingresado a la misma.

Se puede considerar una cola como una estructura ideal e infinita, o bien como una estructura finita.

Operaciones básicas o primitivas para colas infinitas**Crear o New : crea la cola**

Precondiciones : no tiene

Postcondiciones : una cola vacía preparada para ser usada.

Frente o Front : retorna el valor del primer elemento de la cola

Precondiciones : la cola , ya creada, no debe estar vacía.

Postcondiciones: no tiene

Encolar o Enqueue : da de alta en la cola a un elemento (que se ubicará al fondo) pasado por argumento

Precondiciones : la cola debe haber sido creada.

Postcondiciones : la cola modificada con la inserción del nuevo elemento

Desencolar o Dequeue: elimina el elemento del frente de la misma.

Precondiciones : la cola , ya creada, no debe estar vacía.

Postcondiciones : cola modificada por la eliminación del elemento del frente.

Vacia o Empty : devuelve un valor indicando si la cola está vacía.

Precondiciones : la cola debe haber sido creada

Postcondiciones : no tiene

En símbolos:**Primitivas:**

enqueue: $COLA[G] \times G \Rightarrow COLA[G]$

dequeue: $COLA[G] \rightarrow COLA[G]$

front: $COLA[G] \rightarrow G$

empty: $COLA[G] \Rightarrow BOOLEAN$

new: $COLA[G]$

Axiomas :

$\forall x: G, q: COLA[G]$

`empty (new)` (al crear una cola siempre está vacía).

`not empty (enqueue (q, x))`: (si a una cola se le añade un elemento la cola no está vacía).

si `empty (q)`, entonces `front (enqueue (q, x)) = x` (un alta en cola vacía modifica el frente de la misma).

si `not empty (q)` entonces `front (enqueue (q, x)) = front (q)` (alta en cola no vacía no modifica el frente)

si `empty (q)` entonces `dequeue (enqueue (q, x)) = q` (si la cola está vacía y se compone un alta con una baja, la cola quedará en el estado original)

si `not empty (q)` entonces `dequeue (enqueue (q, x)) = enqueue (dequeue (q), x)` (si la cola no está vacía, la composición de un alta de x con una baja es idéntica a la composición de una baja con un alta de x)

Precondiciones:

`Dequeue (q: COLA[G]): not empty (q)`

`Front (q: COLA[G]): not empty (q)`

Operaciones elementales para colas finitas:

A las operaciones expuestas para colas infinitas se le agrega la primitiva

Llena : retorna un valor indicando si la estructura de la cola está llena.

Precondiciones : la cola debe haber sido creada.

Postcondiciones: no tiene.

Además, en la operación Encolar, se modifican las precondiciones.

Precondición de la operación Encolar para colas finitas: la cola debe haber sido creada y no estar llena.

Implementaciones más habituales de las colas:

Utilizando un array 'circular', o bien mediante una lista ligada (con punteros), según se muestra en códigos adjuntos.