

# Programación II

## Tema 3. Colas

Iván Cantador y Rosa Ma Carro

**Cartagena99**

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVIA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- El TAD Cola
- Estructura de datos y primitivas de Cola
- Estructura de datos de Cola como array circular
- Implementación en C de Cola
  - Implementación con front y rear de tipo entero
  
- Anexo
  - Implementación con front y rear de tipo puntero

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue and white background with a subtle wave or cloud-like pattern.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- **El TAD Cola**
- Estructura de datos y primitivas de Cola
- Estructura de datos de Cola como array circular
- Implementación en C de Cola
  - Implementación con front y rear de tipo entero
  
- Anexo
  - Implementación con front y rear de tipo puntero

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue and white background with a subtle wave or cloud-like pattern.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Cola (*queue* en inglés)
  - Colección de elementos FIFO - *First In, First Out*: “el primero que entra, el primero que sale”



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- **Definición de Cola**

- Contenedor de elementos que son insertados y extraídos siguiendo el principio de que el primero que fue insertado será el primero en ser extraído (*FIFO – First In, First Out*)
  - Los elementos se insertan de uno en uno: **insertar**
  - Los elementos se extraen de uno en uno: **extraer**
  - La posición de la cola donde se encuentra el siguiente elemento a ser extraído se denomina **front** (o *head*, inicio)
  - La posición de la cola donde se colocará el siguiente elemento que se inserte se denomina **rear** (o *tail*, fin)

Cartagena99

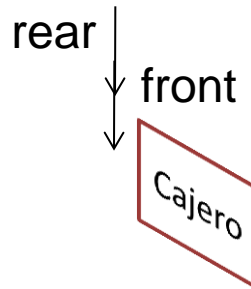
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

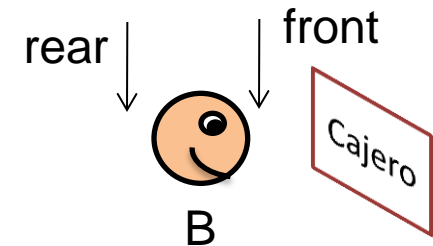
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- **Cola:** contenedor de elementos en el que...
  - la inserción se realiza por un único punto: **rear** / tail / fin
  - la extracción se realiza por un único punto: **front** / head / inicio

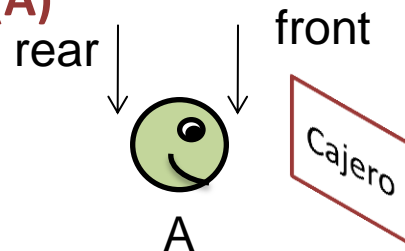
### 1. Cola Vacía



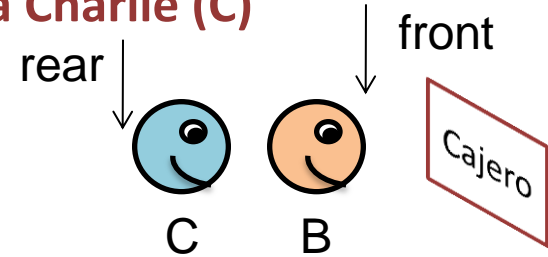
### 4. Sale Alice



### 2. Entra Alice (A)



### 5. Entra Charlie (C)



### 3. Entra Bob (B)

rear

front

### 6. Sale Bob

rear

front

Cartagena99

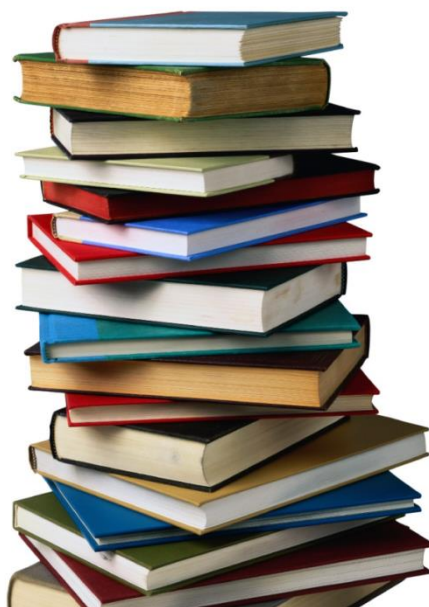
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Diferencias entre los TAD Pila y Cola

- Pila tiene un único punto de entrada y salida; Cola tiene dos
- Pila es LIFO (*Last In, First Out*); Cola es FIFO (*First In, First Out*)



Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE  
LLAMA O ENVIA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Colas en el mundo real: para pagar en comercios, comprar tickets para un espectáculo, sacar dinero de un cajero, ...
  - **Una cola gestiona un acceso concurrente a un único recurso**
- En Informática existen muchos ejemplos de uso de colas
  - Trabajos enviados a impresoras
    - El primer trabajo en llegar es el primero que se imprime: *First Come, First Served* (FCFS)
  - Peticiones a servidores
  - Uso del procesador
    - Un sistema operativo realiza planificaciones de procesos de distintos tipos, gestionando el orden de ejecución de los mismos
- ¡OJO! No todos los elementos tienen que tener siempre

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70



- El TAD Cola
- **Estructura de datos y primitivas de Cola**
- Estructura de datos de Cola como array circular
- Implementación en C de Cola
  - Implementación con front y rear de tipo entero
  
- Anexo
  - Implementación con front y rear de tipo puntero

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue and white background with a subtle wave or cloud-like pattern.

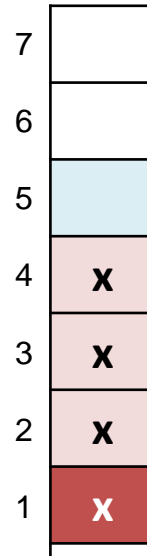
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de datos y primitivas de Cola

- **Una cola está formada por:**
  - **datos:** conjunto de elementos del mismo tipo, ordenados implícitamente y accesibles desde dos puntos: *front* y *rear*
  - **front:** indicador de la posición del próximo elemento a extraer
  - **rear:** indicador de la posición donde colocar el próximo elemento que se inserte



(en este dibujo se asume que la cola tiene tamaño máximo de 8, pero no tiene por qué ser así)

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Primitivas

Cola **cola\_crear**(): crea, inicializa y devuelve una cola

**cola\_liberar**(Cola s): libera (la memoria ocupada por) la cola

boolean **cola\_vacia**(Cola s): devuelve *true* si la cola está vacía y *false* si no

boolean **cola\_llena**(Cola s): devuelve *true* si la cola está llena y *false* si no

status **cola\_insertar**(Cola s, Elemento e): inserta un dato en una cola

Elemento **cola\_extraer**(Cola s): extrae el dato que ocupa el front de la cola

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- El TAD Cola
- Estructura de datos y primitivas de Cola
- **Estructura de datos de Cola como array circular**
- Implementación en C de Cola
  - Implementación con front y rear de tipo entero
  
- Anexo
  - Implementación con front y rear de tipo puntero

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the rest of the text. The logo is set against a background of light blue and orange wavy lines.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de datos de Cola como array circular<sup>12</sup>

- Ejemplo de ejecución de operaciones en una cola

1) cola\_inicializar(q)

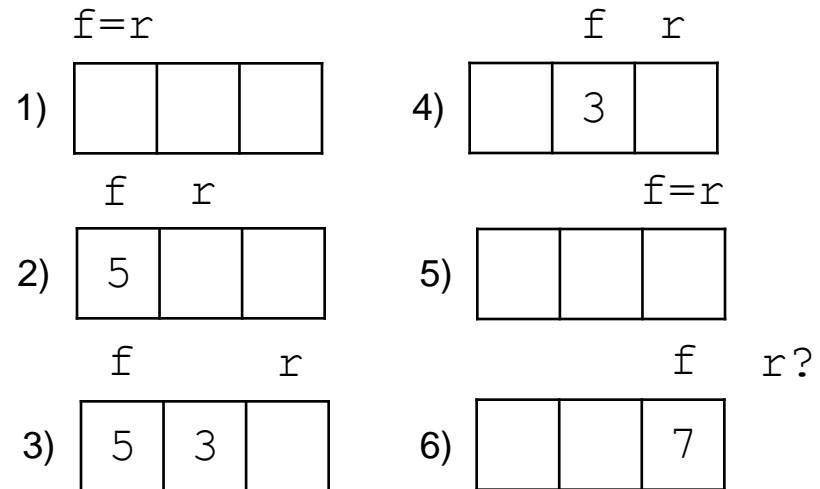
2) cola\_insertar(q, 5)

3) cola\_insertar(q, 3)

4) cola\_extraer(q)

5) cola\_extraer(q)

6) cola\_insertar(q, 7)



- Problemas

- Limitación del número máximo de elementos

Cartagena99

CLASES PARTICULARES, TUTORIAS TECNICAS ONLINE  
LLAMA O ENVIA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de datos de Cola como array circular<sup>13</sup>

- Soluciones al desperdicio de espacio
  - 1) Cada vez que se extrae un elemento, se desplazan todos los datos una posición en el array
    - Ineficiente
  - 2) Cuando *rear* llega al final del array, se desplazan todos los elementos una posición en el array
    - (menos) Ineficiente
  - 3) Implementación de la cola como un **array circular**
    - Más eficiente

Cartagena99

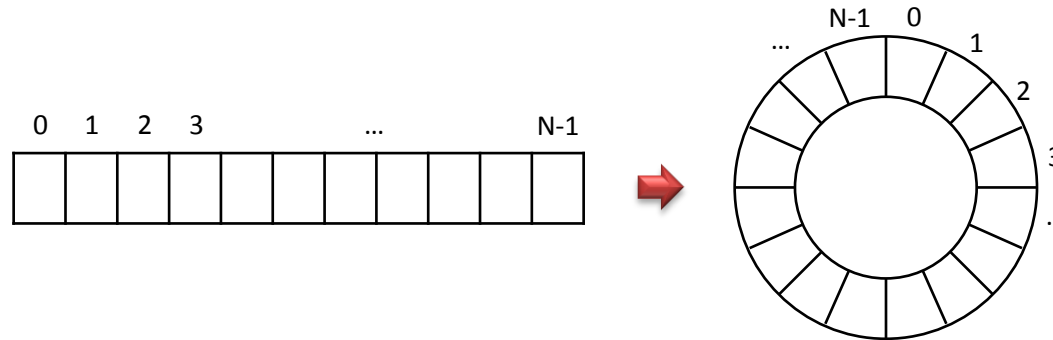
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de datos de Cola como array circular<sup>14</sup>

- Cola circular



- ¿Cómo implementarla?

- Incrementando *front* y *rear* módulo COLA\_MAX

`front = (front+1) % COLA_MAX`

`rear = (rear+1) % COLA_MAX`

- Problema vigente

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

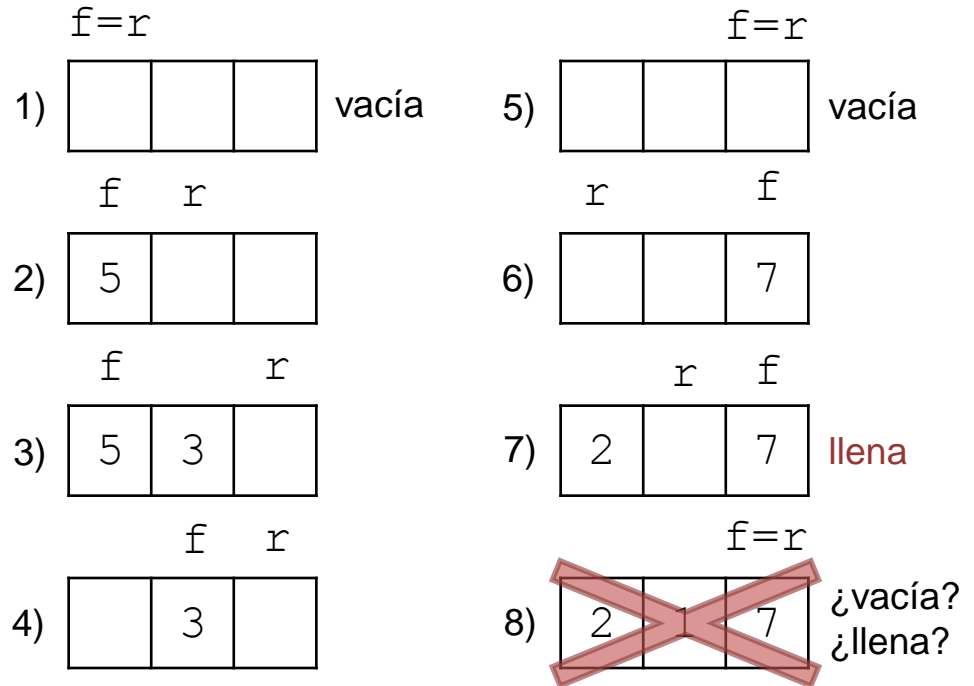
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Estructura de datos de Cola como array circular<sup>15</sup>

## • Ejemplo de ejecución de operaciones

- 1) cola\_inicializar(q)
- 2) cola\_insertar(q, 5)
- 3) cola\_insertar(q, 3)
- 4) cola\_extraer(q, e)
- 5) cola\_extraer(q, e)
- 6) cola\_insertar(q, 7)
- 7) cola\_insertar(q, 2)
- 8) cola\_insertar(q, 1)



## • Conflicto cola llena/vacía

- front == rear → ¿Cola vacía o llena?
- **Solución:** sacrificar un hueco libre en el array

Prohibir la inserción cuando sólo queda un hueco.

Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

Programación II – Tema 3: Colas



- El TAD Cola
- Estructura de datos y primitivas de Cola
- Estructura de datos de Cola como array circular
- **Implementación en C de Cola**
  - **Implementación con front y rear de tipo entero**
- Anexo
  - Implementación con front y rear de tipo puntero

The logo for Cartagena99 features the text 'Cartagena99' in a stylized, green, serif font. The '99' is significantly larger and more prominent than the 'Cartagena' part. The text is set against a light blue and white background with a subtle wave-like pattern.

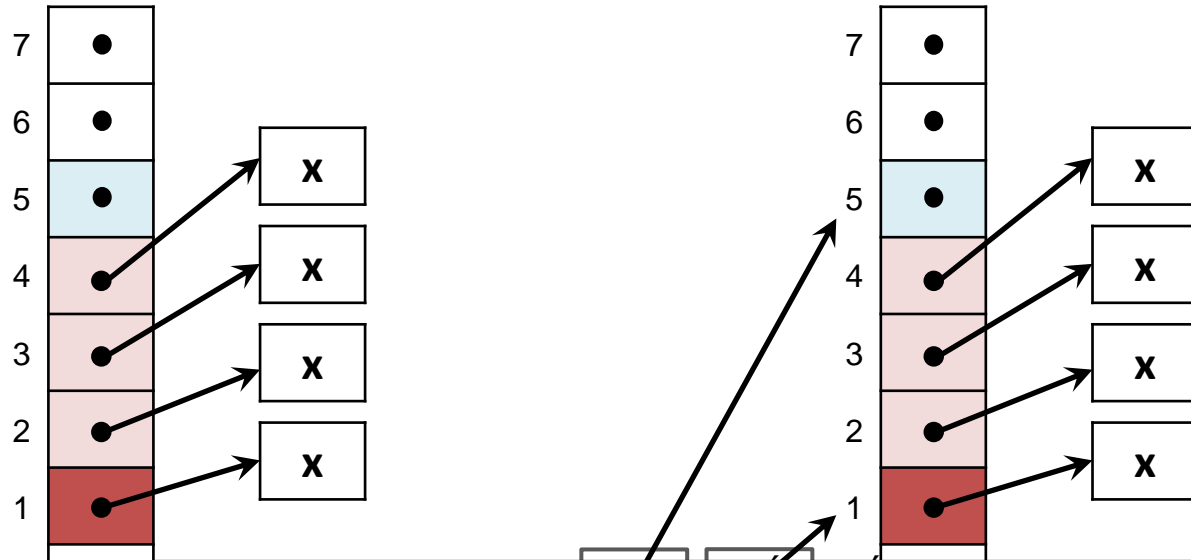
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • EdD en C

- **datos**: en este tema será un array de punteros: `Elemento *datos[];`
- **front, rear**: en este tema se declarará de 2 maneras (versiones) distintas
  - Como enteros: `int front, rear;`
  - Como punteros a elemento del array: `Elemento **front, **rear;`



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

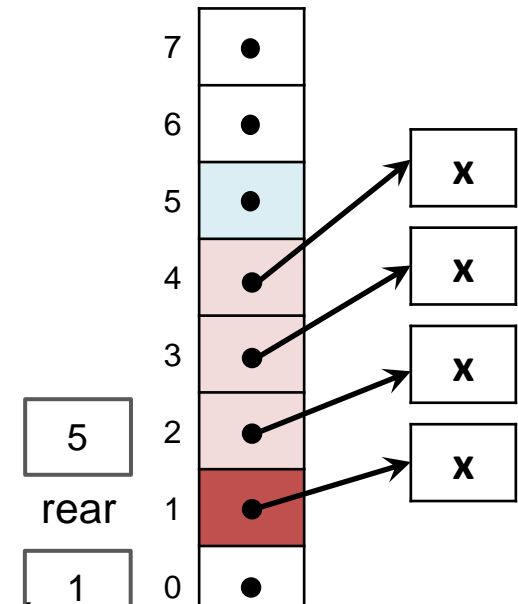
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo entero
  - Se asume la existencia del TAD Elemento
  - EdD de Cola mediante un array

```
// En cola.h
typedef struct _Cola Cola;

// En cola.c
#define COLA_MAX 8
struct _Cola {
    Elemento *datos[COLA_MAX];
    int front; // Primer elemento
    int rear; // Primer hueco tras
              // el último elemento
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo entero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

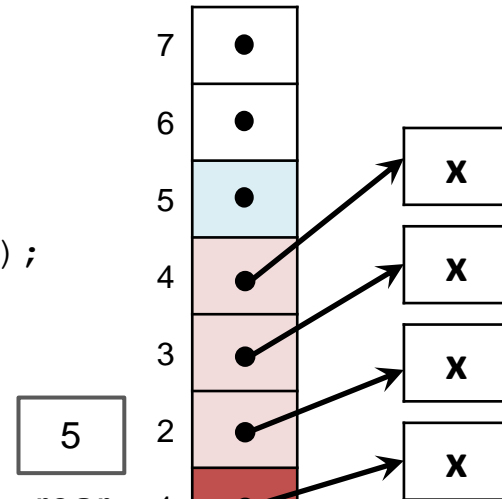
```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```

## • Estructura de datos (en cola.c)

```
struct _Cola {
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

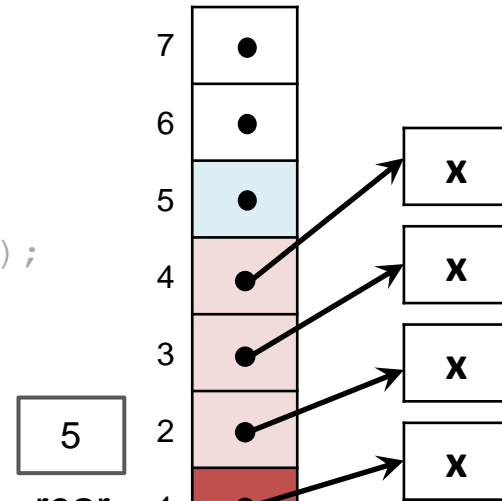
## • Implementación con front y rear de tipo entero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

Cartagena99

# Implementación en C de Cola

- Implementación con front y rear de tipo entero

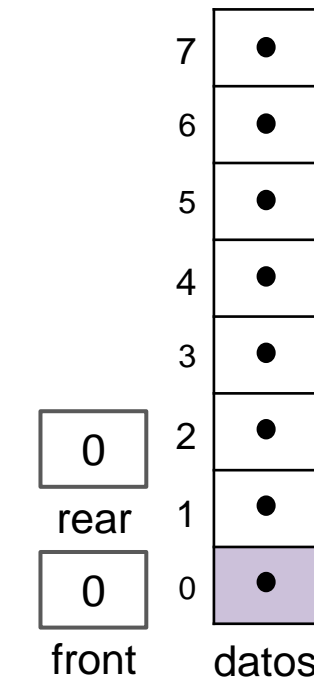
```
Cola *cola_crear() {
    Cola *pq = NULL;
    int i;

    pq = (Cola *) malloc(sizeof(Cola));
    if (pq==NULL) {
        return NULL;
    }

    pq->front = 0;
    pq->rear = 0;

    return pq;
}
```

vacía



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

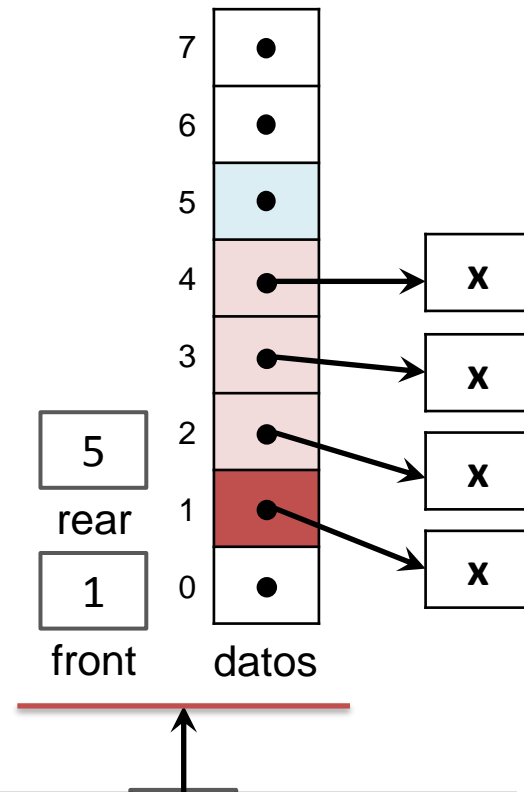
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo entero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
```

```
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

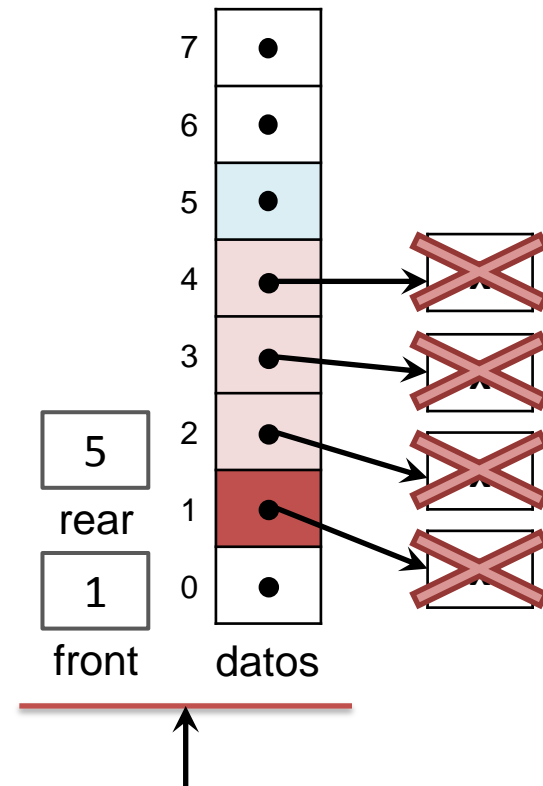
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo entero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    int i;

    if (pq!=NULL) {
        i = pq->front;
        while (i!=pq->rear) {
            elemento_liberar(pq->datos[i]);
            i = (i+1) % PILA_MAX;
        }
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70



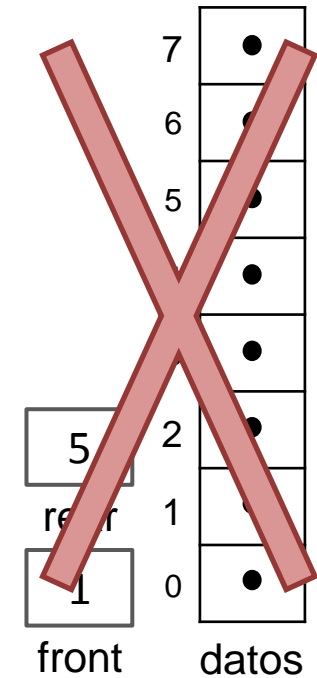
# Implementación en C de Cola

## • Implementación con front y rear de tipo entero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    int i;

    if (pq!=NULL) {
        i = pq->front;
        while (i!=pq->rear) {
            elemento_liberar(pq->datos[i]);
            i = (i+1) % PILA_MAX;
        }
        free(pq);
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

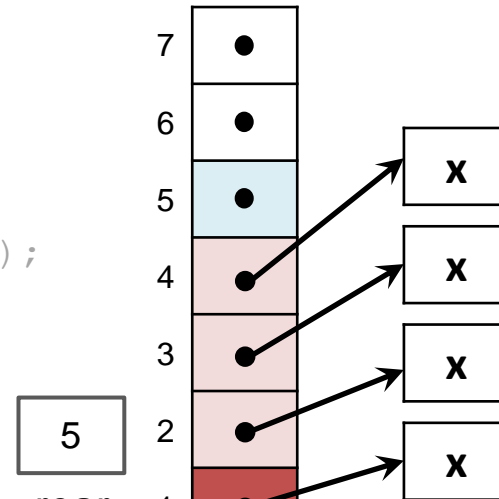
## • Implementación con front y rear de tipo entero

- Asumimos la existencia del **TAD Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

Cartagena99

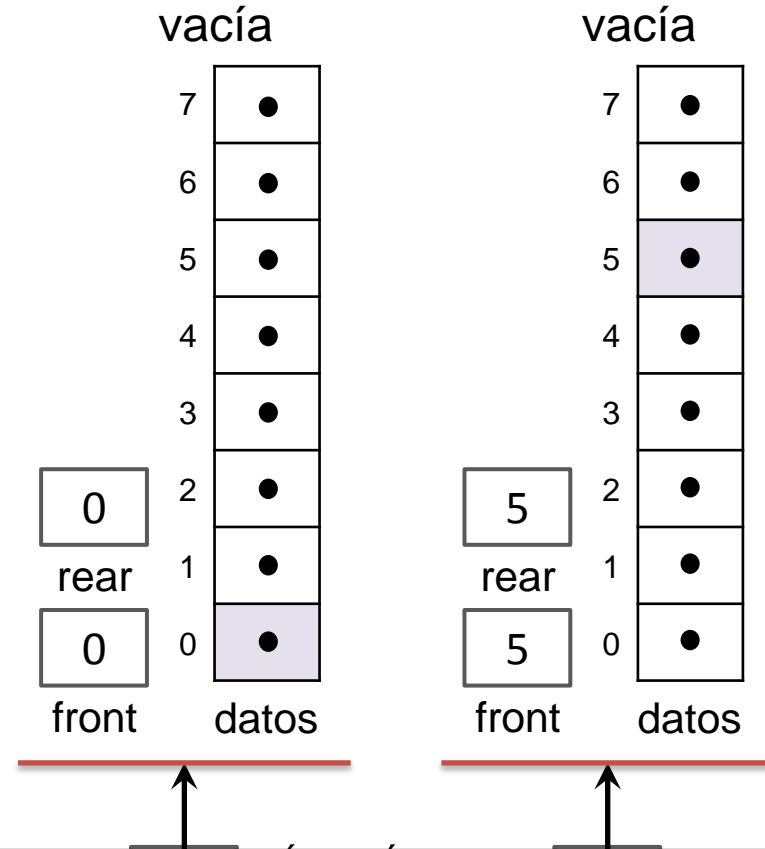
# Implementación en C de Cola

- Implementación con front y rear de tipo entero

```
boolean cola_vacia(const Cola *pq) {
    if (pq == NULL) {
        return NULL_BOOLEAN;
    }

    if (pq->rear == pq->front) {
        return TRUE;
    }

    return FALSE;
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

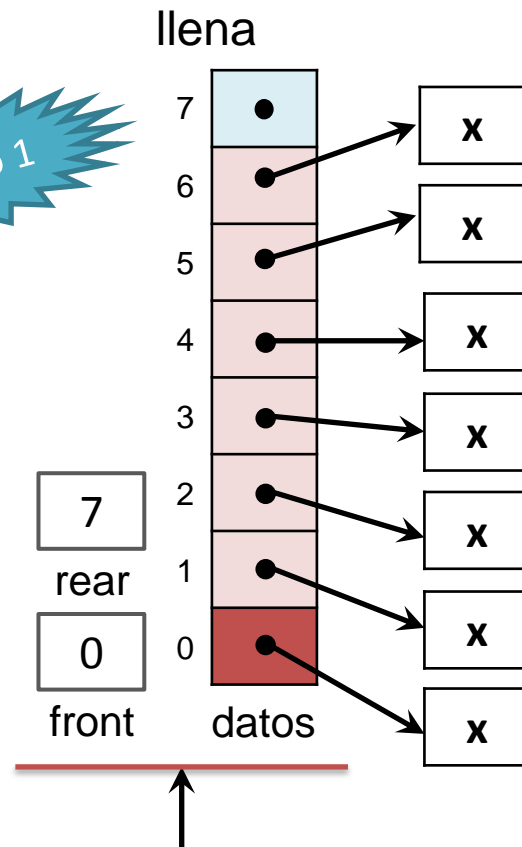
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo entero

```
boolean cola_llena(const Cola *pq) {  
    if (pq == NULL) {  
        return NULL_BOOLEAN;  
    }  
  
    if ((pq->rear+1)%COLA_MAX == pq->front) {  
        return TRUE;  
    }  
    return FALSE;  
}
```

Ejemplo 1



Cartagena99

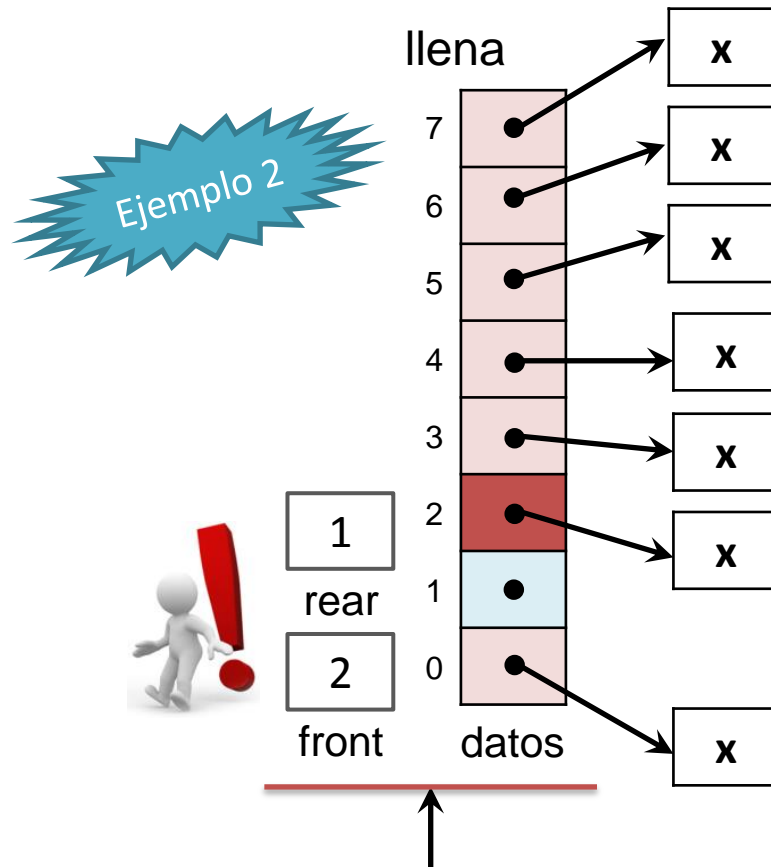
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo entero

```
boolean cola_llena(const Cola *pq) {  
    if (pq == NULL) {  
        return NULL_BOOLEAN;  
    }  
  
    if ((pq->rear+1)%COLA_MAX == pq->front) {  
        return TRUE;  
    }  
    return FALSE;  
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo entero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

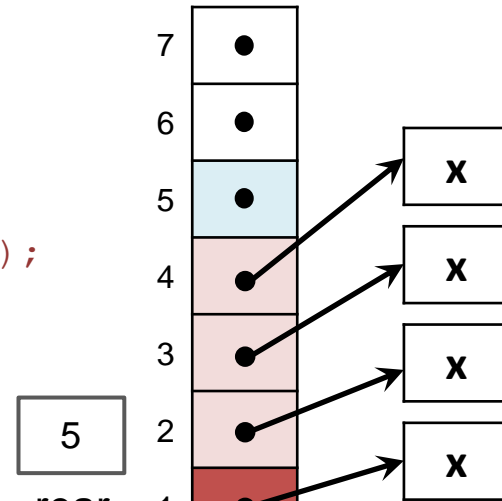
## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```



Cartagena99

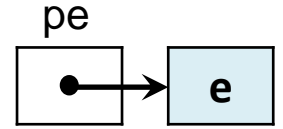
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

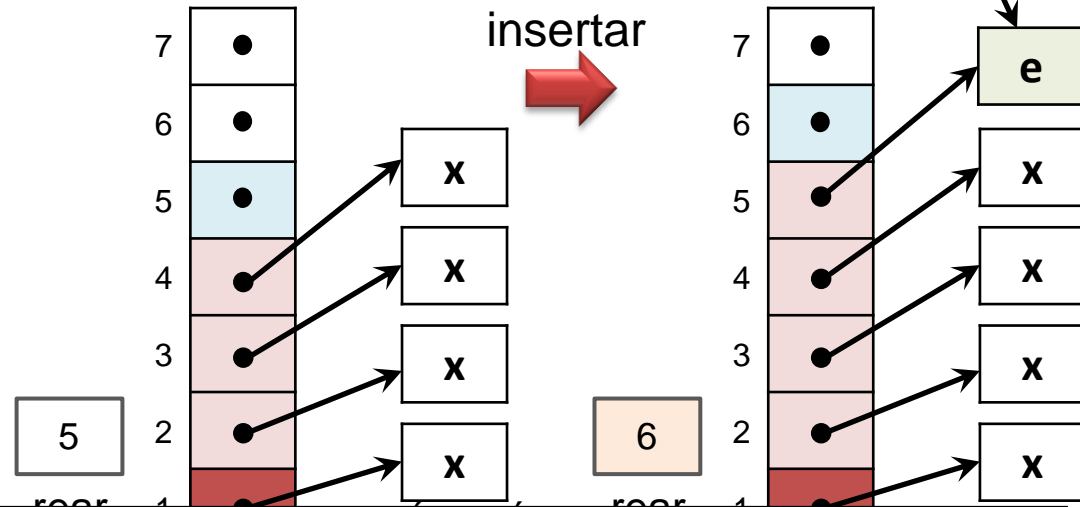
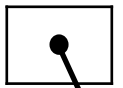
# Implementación en C de Cola

- Implementación con front y rear de tipo entero

```
status cola_insertar(Cola *pq, const Elemento *pe) {
```



aux



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo entero

```

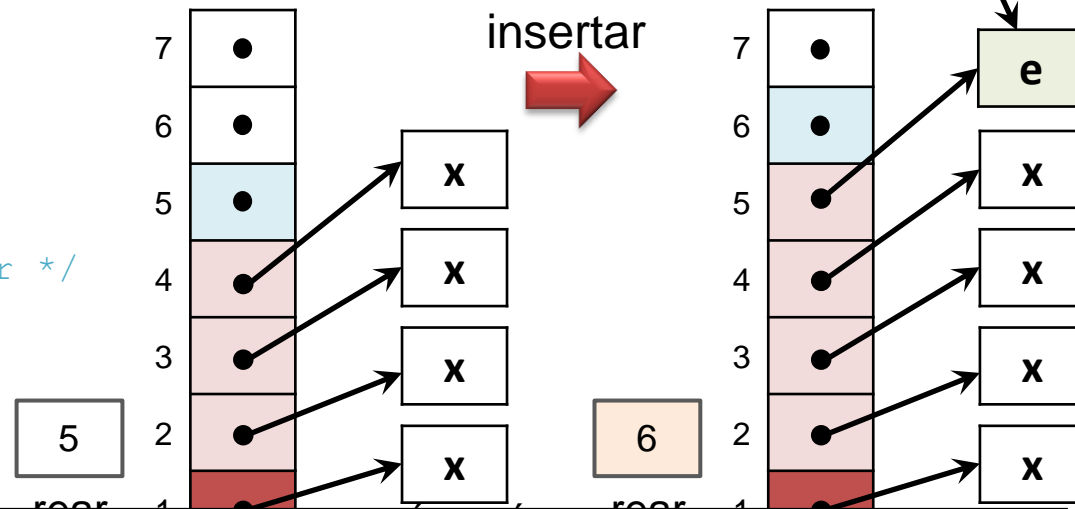
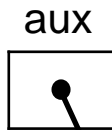
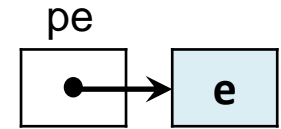
status cola_insertar(Cola *pq, const Elemento *pe) {
    Elemento *aux = NULL;

    if (pq == NULL || pe == NULL || cola_llena(pq) == TRUE) {
        return ERROR;
    }

    aux = elemento_copiar(pe);
    if (aux == NULL) {
        return ERROR;
    }

    /* Guardamos el dato en el rear */
    pq->datos[pq->rear] = aux;

    /* Actualizamos el rear */
    pq->rear = (pq->rear+1)%COLA_MAX;
  
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70



# Implementación en C de Cola

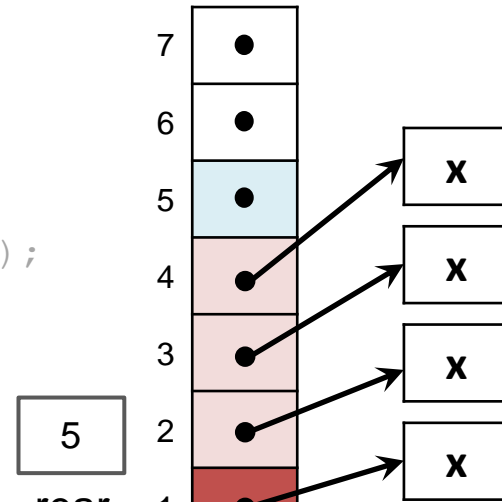
## • Implementación con front y rear de tipo entero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

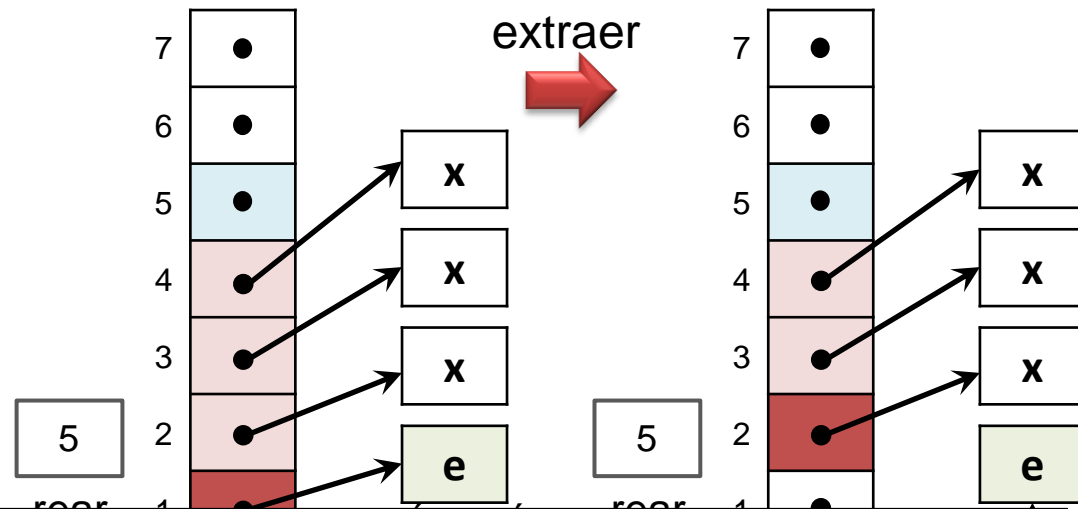
Cartagena99

# Implementación en C de Cola

- Implementación con front y rear de tipo entero

```
Elemento *cola_extraer(Cola *pq) {
```

```
}
```



CLASES PARTICULARES, TUTORIAS TÉCNICAS ONLINE  
LLAMA O ENVIA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo entero

```

Elemento *cola_extraer(Cola *pq) {
    Elemento *pe = NULL;

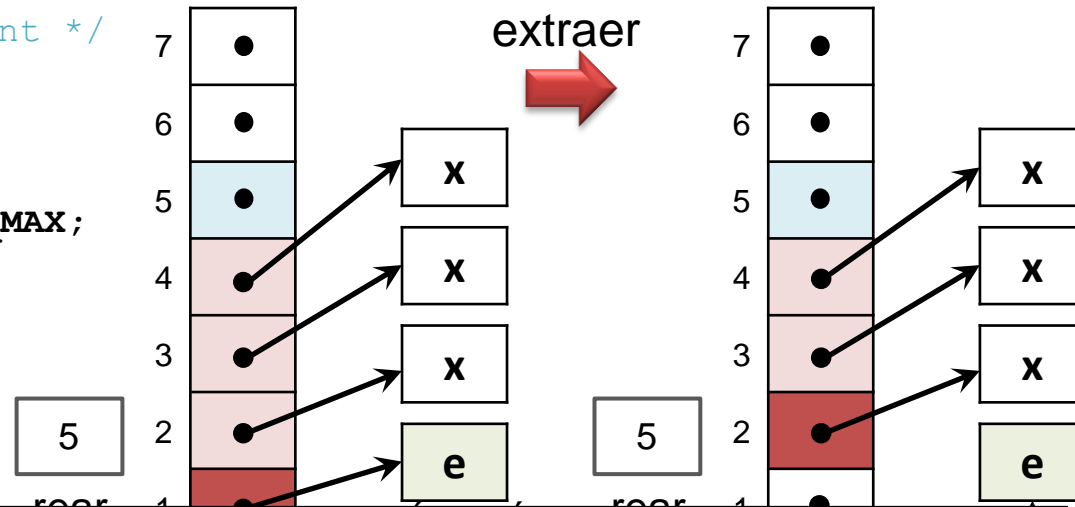
    if (pq == NULL || cola_vacia(pq) == TRUE) {
        return NULL;
    }

    /* Recuperamos el dato del front */
    pe = pq->datos[pq->front];

    /* Actualizamos el front */
    pq->front = (pq->front+1) % COLA_MAX;

    return pe;
}

```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
 CALL OR WHATSAPP: 689 45 44 70

- El TAD Cola
- Estructura de datos y primitivas de Cola
- Estructura de datos de Cola como array circular
- Implementación en C de Cola
  - Implementación con front y rear de tipo entero
  
- **Anexo**
  - **Implementación con front y rear de tipo puntero**

The logo for Cartagena99, featuring the text 'Cartagena99' in a stylized font with a blue and orange gradient background.

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

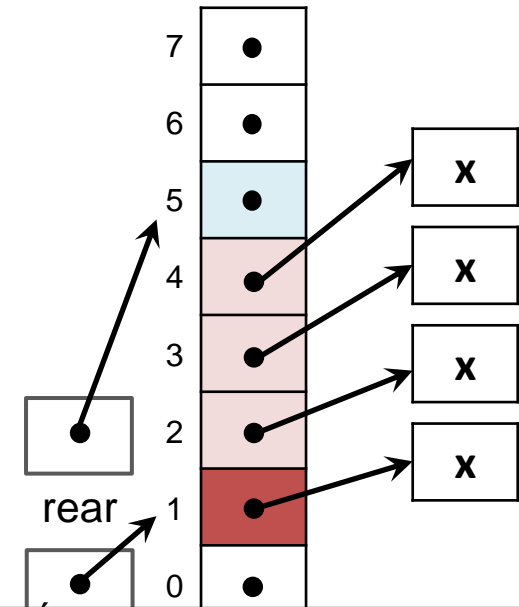
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo puntero
  - Se asume la existencia del TAD Elemento
  - EdD de Cola mediante un array

```
// En cola.h
typedef struct _Cola Cola;

// En cola.c
#define COLA_MAX 8
struct _Cola {
    Elemento *datos[COLA_MAX];
    Elemento **front; // Primer elemento
    Elemento **rear;  // Ultimo elemento
};
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

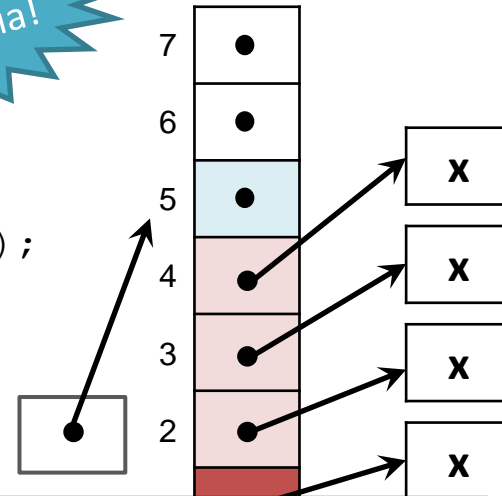
```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```

## • Estructura de datos (en cola.c)

```
struct _Cola {
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

# Implementación en C de Cola

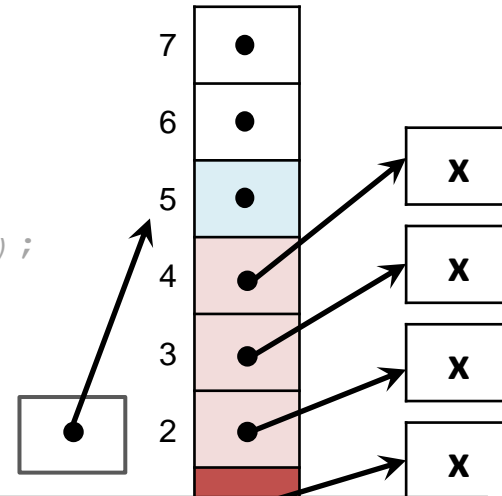
## • Implementación con front y rear de tipo puntero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

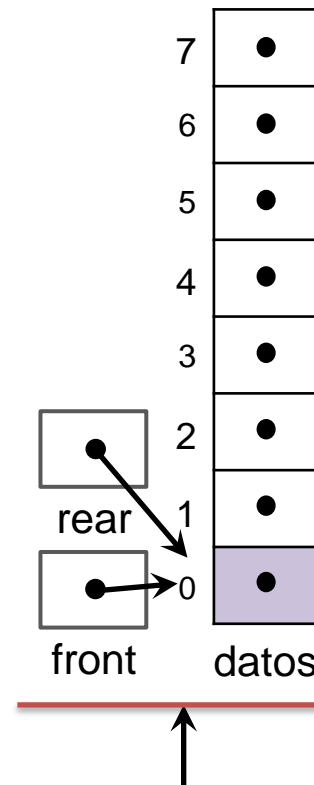
```
Cola *cola_crear() {
    Cola *pq = NULL;
    int i;

    pq = (Cola *) malloc(sizeof(Cola));
    if (pq==NULL) {
        return NULL;
    }

    pq->rear = &(pq->datos[0]); // pq->rear = pq->datos
    pq->front = &(pq->datos[0]); // pq->front = pq->datos

    return pq;
}
```

vacía



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70



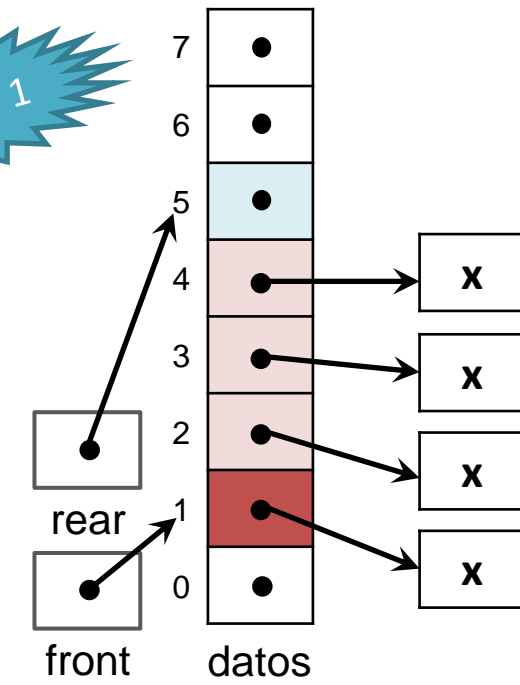
# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
```

Ejemplo 1



Cartagena99

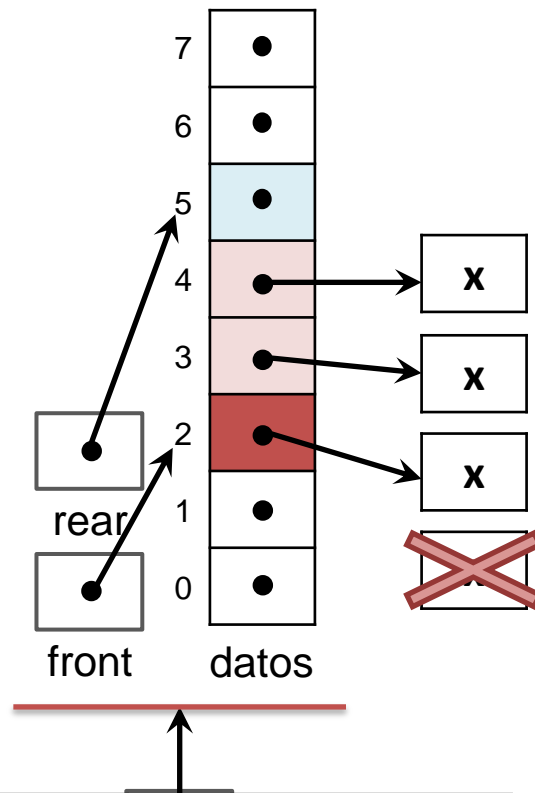
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70  
---  
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

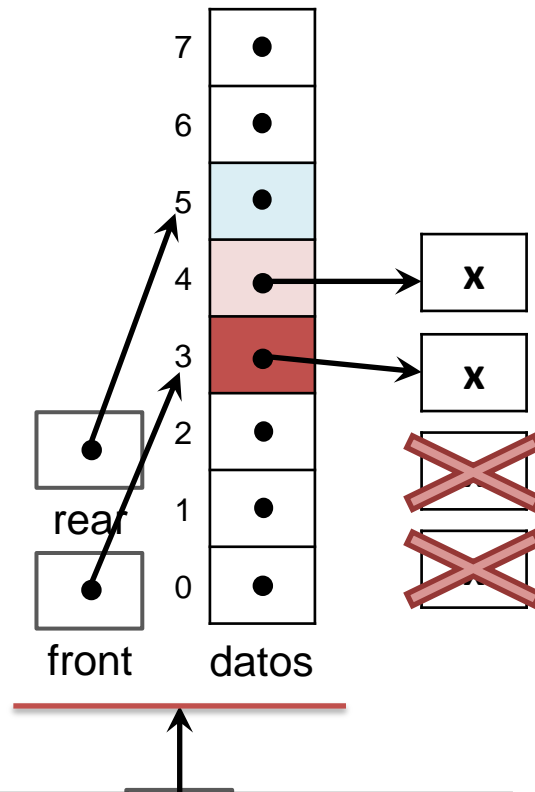
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

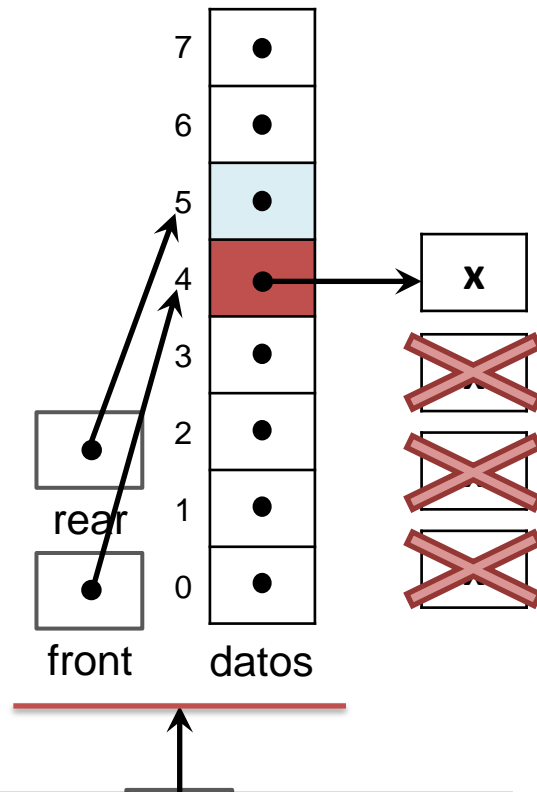
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {  
    if (pq != NULL) {  
        while (pq->front != pq->rear) {  
            elemento_liberar(* (pq->front));  
            if (pq->front != &(pq->datos[COLA_MAX-1])) {  
                pq->front = pq->front+1;  
            }  
            else {  
                pq->front = &(pq->datos[0]);  
            }  
        }  
    }  
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

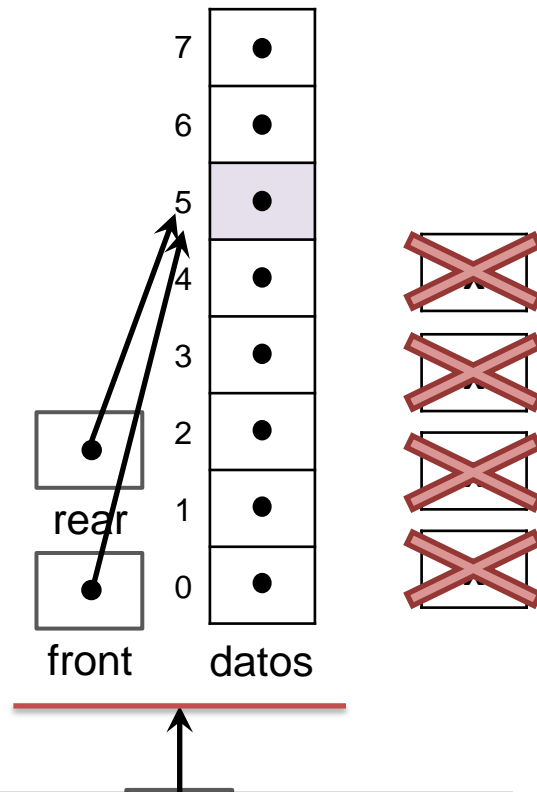
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

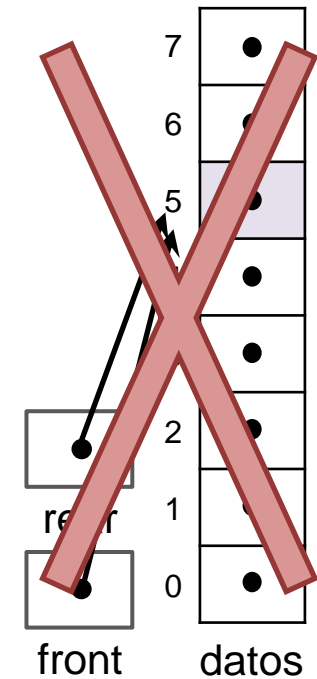
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(*(pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

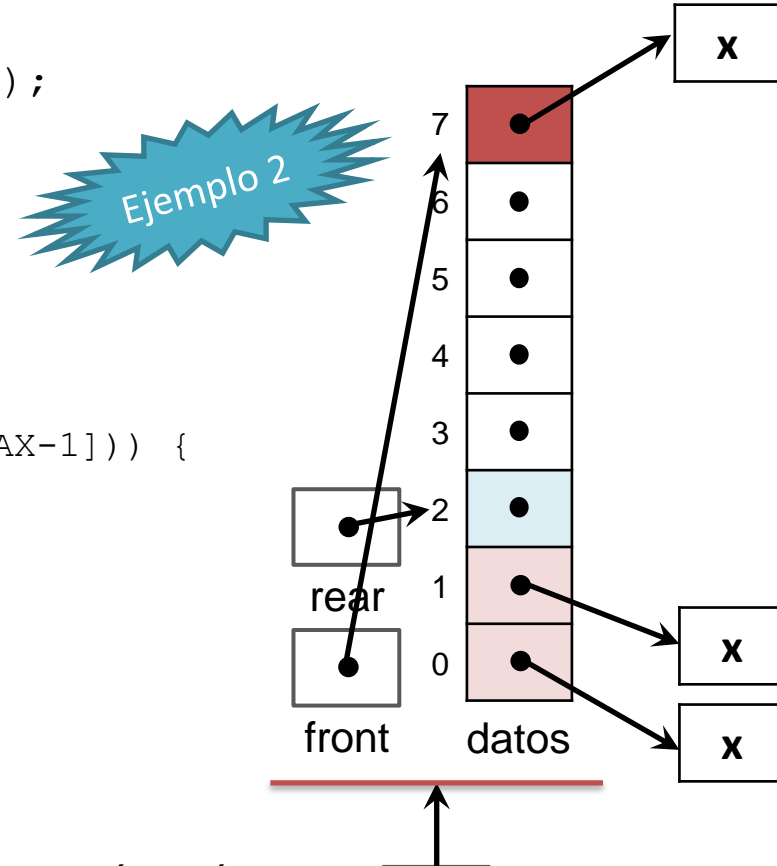
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

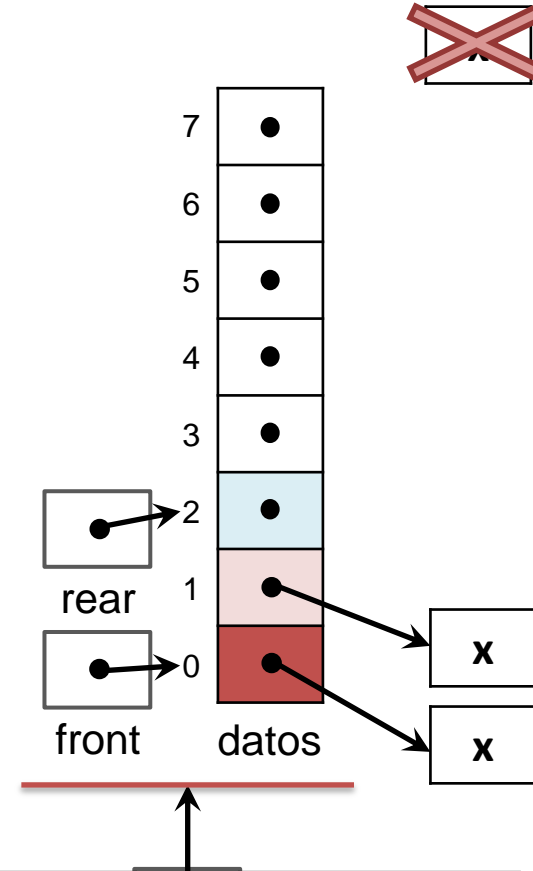
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP: 689 45 44 70

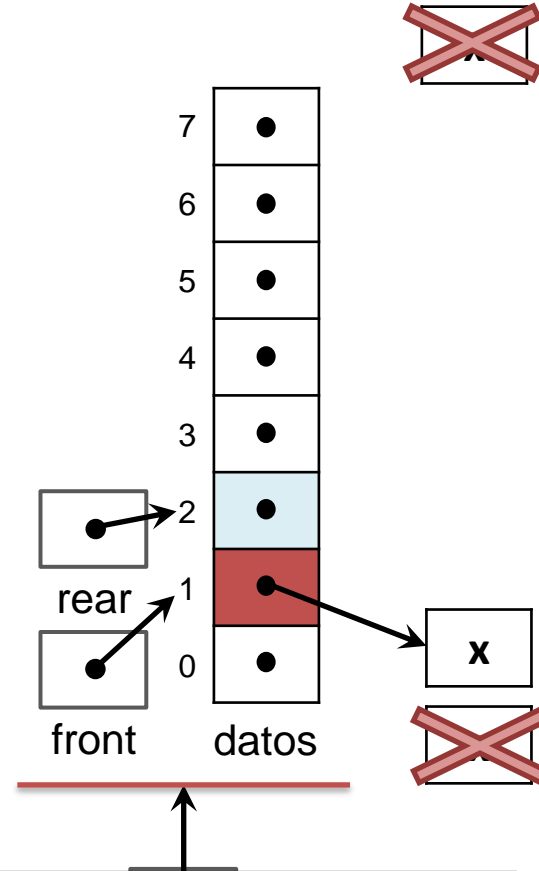


# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

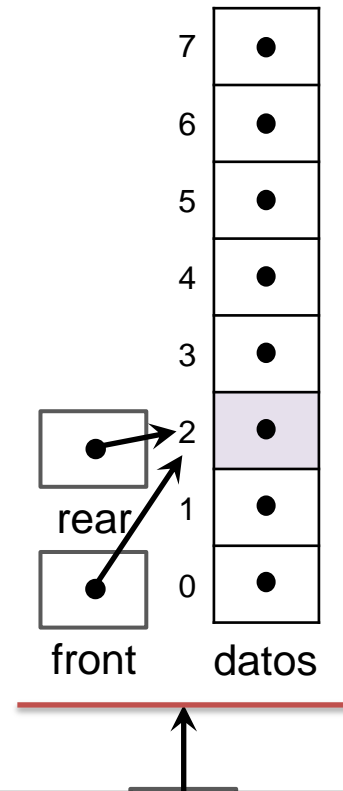
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(* (pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

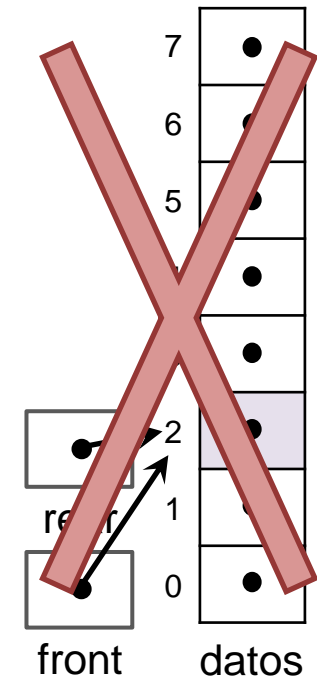
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

## • Implementación con front y rear de tipo puntero

Existe: `void elemento_liberar(Elemento *pe);`

```
void cola_liberar(Cola *pq) {
    if (pq != NULL) {
        while (pq->front != pq->rear) {
            elemento_liberar(*(pq->front));
            if (pq->front != &(pq->datos[COLA_MAX-1])) {
                pq->front = pq->front+1;
            }
            else {
                pq->front = &(pq->datos[0]);
            }
        }
        free(pq);
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

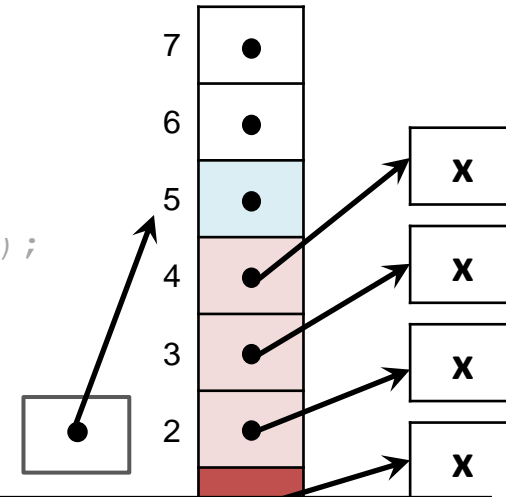
## • Implementación con front y rear de tipo puntero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

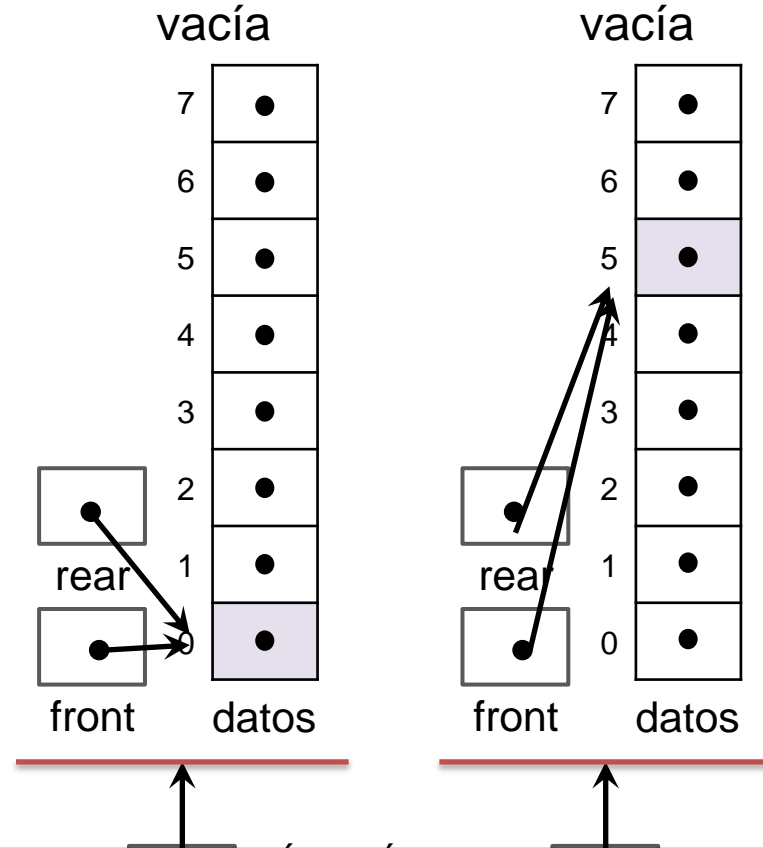
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

- Implementación con front y rear de tipo puntero

```
boolean cola_vacia(const Cola *pq) {
    if (pq == NULL) {
        return NULL_BOOLEAN;
    }

    if (pq->rear == pc->front) {
        return TRUE;
    }
    return FALSE;
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Implementación con front y rear de tipo puntero

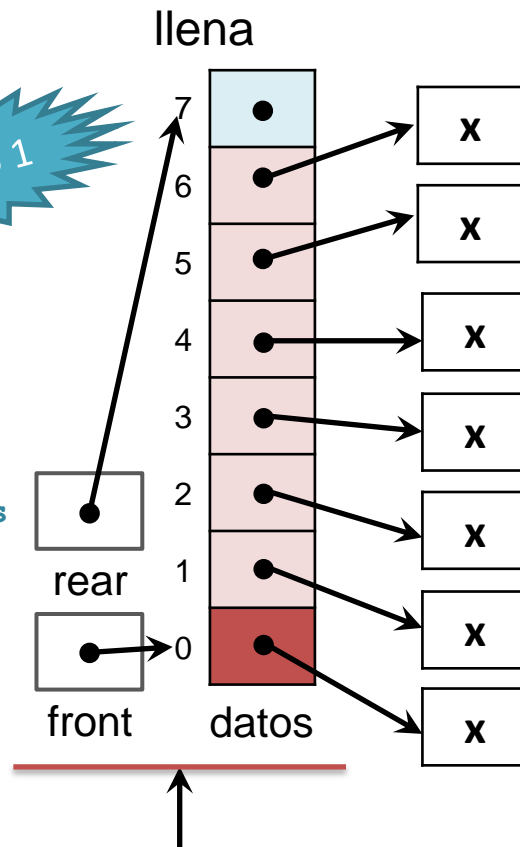
```
boolean cola_llena(const Cola *pq) {
    Elemento **aux = NULL;

    if (pq == NULL) {
        return NULL_BOOLEAN;
    }

    // Apuntamos aux donde avanzaria rear si se incrementase
    if (pq->rear == &(pq->datos[COLA_MAX-1])) {
        aux = &(pq->datos[0]); // Al comienzo del array.
    } // Equivale a aux = pq->datos
    else {
        aux = pq->rear + 1; // A la siguiente posicion
    }

    // Si aux (que es rear+1) coincide con front, cola llena
    if (aux == pq->front) {
        return TRUE;
    }
}
```

Ejemplo 1



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Implementación con front y rear de tipo puntero

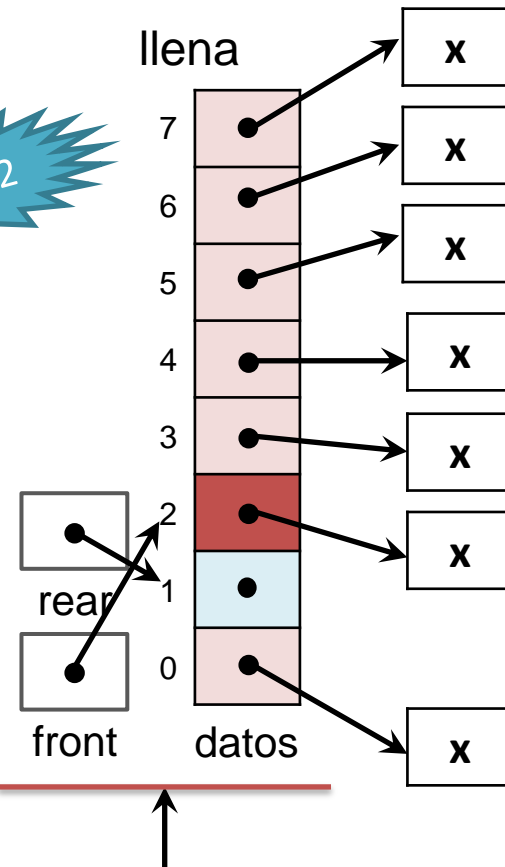
```
boolean cola_llena(const Cola *pq) {
    Elemento **aux = NULL;

    if (pq == NULL) {
        return NULL_BOOLEAN;
    }

    // Apuntamos aux donde avanzaria rear si se incrementase
    if (pq->rear == &(pq->datos[COLA_MAX-1])) {
        aux = &(pq->datos[0]); // Al comienzo del array.
    } // Equivale a aux = pq->datos
    else {
        aux = pc->rear + 1; // A la siguiente posicion
    }

    // Si aux (que es rear+1) coincide con front, cola llena
    if (aux == pc->front) {
```

Ejemplo 2



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

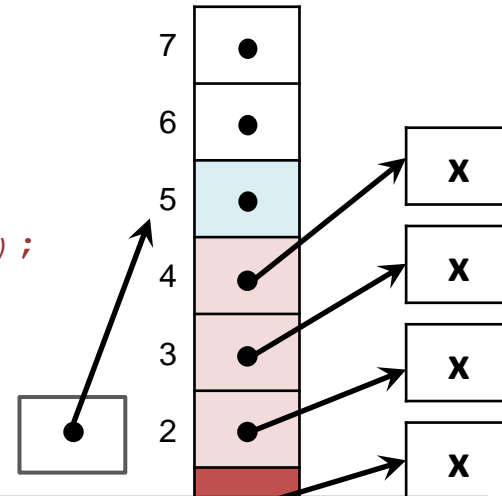
## • Implementación con front y rear de tipo puntero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

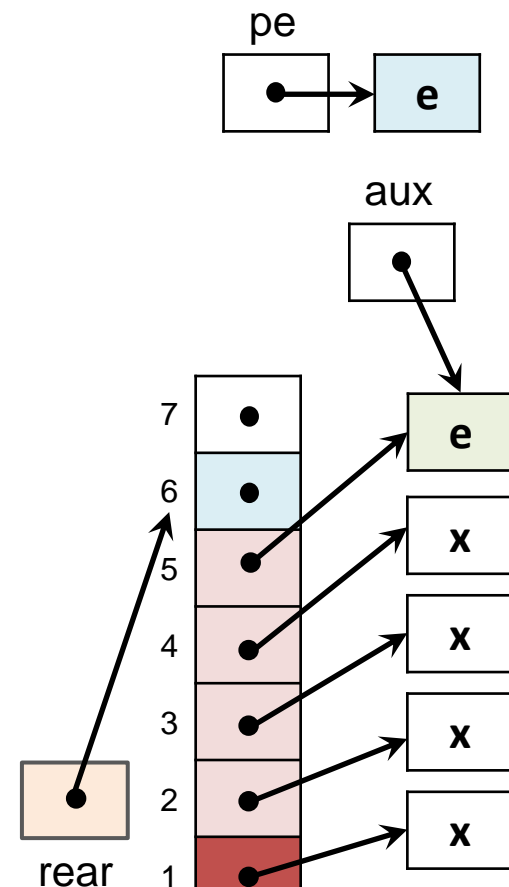
---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70



- Implementación con front y rear de tipo puntero

```
status cola_insertar(Cola *pq, const Elemento *pe){
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

## • Implementación con front y rear de tipo puntero

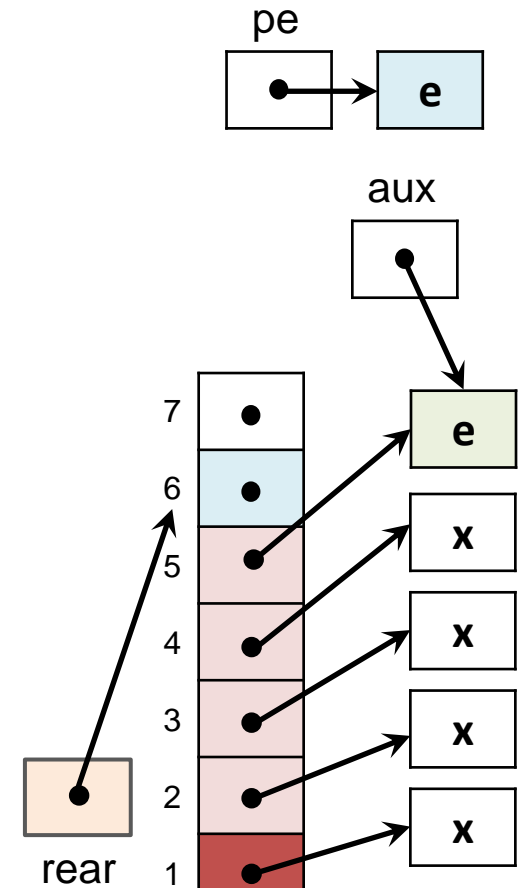
```
status cola_insertar(Cola *pq, const Elemento *pe){
    Elemento *aux = NULL;

    if (pq == NULL || pe == NULL || cola_llena(pq) == TRUE) {
        return ERROR;
    }

    aux = elemento_copiar(pe);
    if (aux == NULL) {
        return ERROR;
    }

    /* Guardamos el dato en el rear */
    *(pq->rear) = aux;

    /* Actualizamos el rear */
    if (pq->rear == &(pq->datos[COLA_MAX-1])) {
        pq->rear = &(pq->datos[0]); // pq->rear = pq->datos
    }
    else {
        pq->rear++;
    }
}
```



Cartagena99

CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

# Implementación en C de Cola

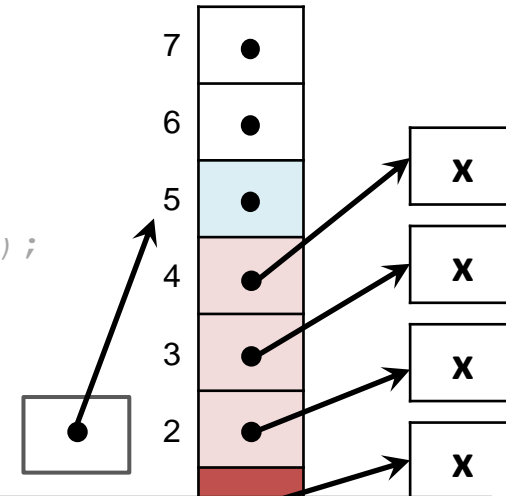
## • Implementación con front y rear de tipo puntero

- Asumimos la existencia del TAD **Elemento** que, entre otras, tiene asociadas las primitivas *liberar* y *copiar*:

```
void elemento_liberar(Elemento *pe);
Elemento *elemento_copiar(const Elemento *pe);
```

## • Primitivas (prototipos en cola.h)

```
Cola *cola_crear();
void cola_liberar(Cola *pq);
boolean cola_vacia(const Cola *pq);
boolean cola_llena(const Cola *pq);
status cola_insertar(Cola *pq, const Elemento *pe);
Elemento *cola_extraer(Cola *pq);
```



## • Estructura de datos (en cola.c)

```
struct _Cola {
```

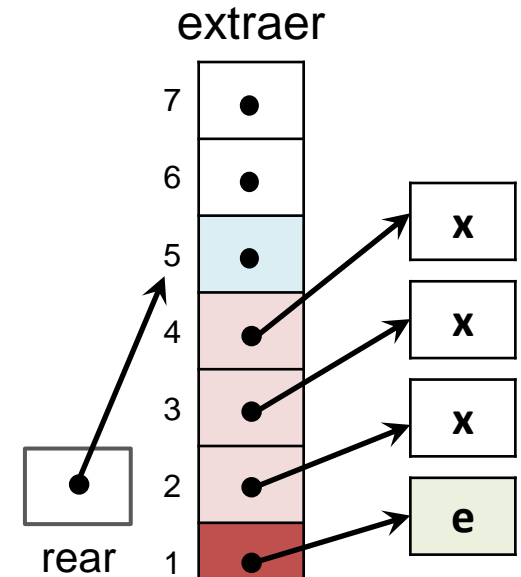
CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

- Implementación con front y rear de tipo puntero

```
Elemento *cola_extraer(const Cola *pq){
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70

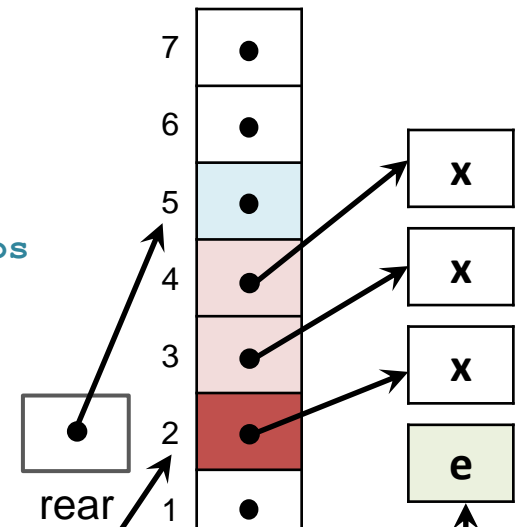
## • Implementación con front y rear de tipo puntero

```
Elemento *cola_extraer(const Cola *pq){
    Elemento *pe = NULL;

    if (pq == NULL || cola_vacia(pq) == TRUE) {
        return NULL;
    }

    /* Recuperamos el dato del front */
    pe = *(pq->front);

    /* Actualizamos el front */
    if (pq->front == &(pq->datos[COLA_MAX-1])) {
        pq->front = &(pq->datos[0]); // pq->front= pq->datos
    }
    else {
        pq->front++;
    }
}
```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE  
LLAMA O ENVÍA WHATSAPP: 689 45 44 70

---  
ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS  
CALL OR WHATSAPP:689 45 44 70