

## Colas dinámicas

Sofía Beatriz Pérez  
Daniel Agustín Rosso

`sperez@iua.edu.ar`

`drosso@iua.edu.ar`

Centro Regional Univesitario Córdoba  
Instituto Univeristario Areonáutico

Clase número 8 - Ciclo lectivo 2023

# Agenda

Colas dinámicas

Colas dinámicas: operaciones

## Disclaimer

Los siguientes slides tienen el objetivo de dar soporte al dictado de la asignatura. De ninguna manera pueden sustituir los apuntes tomados en clases y/o la asistencia a las mismas.

Es importante mencionar que todos este material se encuentra en un proceso de mejora continua.

Si encuentra bugs, errores de ortografía o redacción, por favor repórtelo a [sperez@iua.edu.ar](mailto:sperez@iua.edu.ar) y/o [drosso@iua.edu.ar](mailto:drosso@iua.edu.ar). También puede abrir issues en el repositorio de este link: [▶ infoI\\_IUA\\_GitLab](#)

# Colas dinámicas: introducción

Una cola de espera es similar a la fila de pagos en una caja de un supermercado. La primera persona en la línea es atendida primero y los otros clientes ingresan por la parte final y esperan ser atendidos. Los nodos de la cola son solamente eliminados por la **parte delantera** y son incluidos o insertos únicamente or **la parte trasera**.

Este tipo de dato también es conocido por sus siglas en ingles FIFO<sup>1</sup>.

---

<sup>1</sup>First Input First Output.

## Colas dinámicas: aplicaciones

- Colas de impresion
- Sincronización de procesos de forma asíncrona: uso de la FIFO para sincronizar
- Manejo de interrupciones en sistemas de tiempo real
- Sistemas de esperas en Call Centers

## Colas dinámicas: gráfico

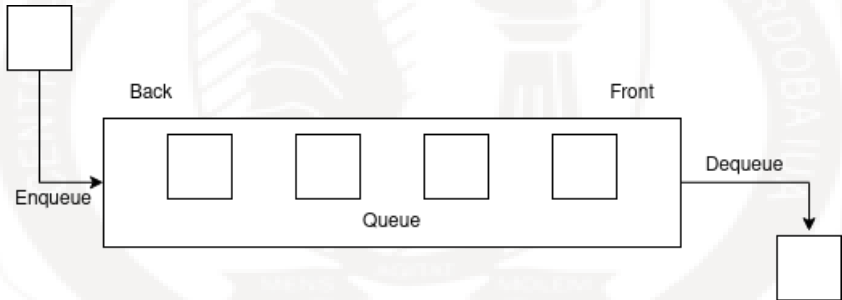


Figure: Representación gráfica de una cola.

## Colas dinámicas: operaciones

Las operaciones más comunes con colas dinámicas son:

- Agregar un nodo
- Eliminar un nodo
- Calcular la cantidad de elementos almacenados en la FIFO
- Imprimir todos los nodos de la FIFO

## Colas dinámicas: creación

```
1 struct node
2 {
3     int data;
4     struct node *link;
5 };
```

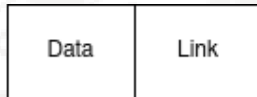


Figure: Representación gráfica de una cola.



## Colas dinámicas: punteros a front y back

```
1 struct node *front=NULL;  
2 struct node *back = NULL;
```

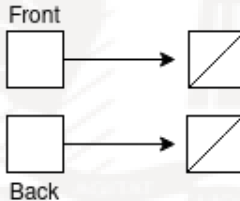
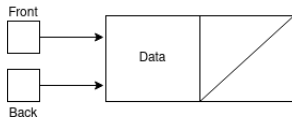


Figure: Punteros a la estructura front y rear de la queue.



## Colas dinámicas: push del primer nodo

```
1 new_node=(struct node *)malloc(sizeof(struct node));
2
3 new_node->data = value;
4 new_node->link=NULL;
5
6 if(back == NULL)
7 {
8     back = new_node;
9     front = back;
10 }
11 else
12 {
13     back->link = new_node;
14     back = new_node;
15 }
```

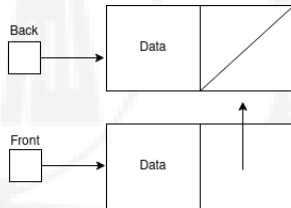


Push del primer nodo



## Colas dinámicas: push de los nodos siguientes

```
1 new_node=(struct node *)malloc(sizeof(struct node));
2
3 new_node->data = value;
4 new_node->link=NULL;
5
6 if (back == NULL)
7 {
8     back = new_node;
9     front = back;
10 }
11 else
12 {
13     back->link = new_node;
14     back = new_node;
15 }
```



Push del primer nodo

## Colas dinámicas: push de los nodos siguientes

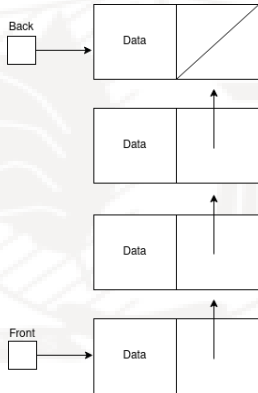
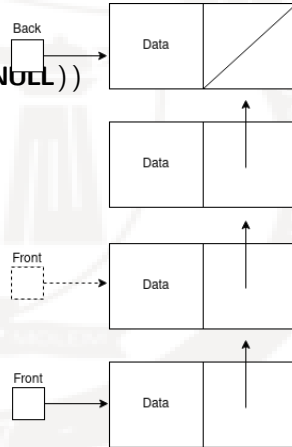


Figure: Representación gráfica de una cola.

## Colas dinámicas: pop de nodos

```
1 if ((front == back) && (back == NULL))
2 {
3     printf("Vacía");
4     exit(0);
5 }
6 temp = front;
7 front = (front) -> link;
8 if (front == NULL)
9     rear = NULL;
10 free(temp);
11 }
```



Pop de un nodo

## Colas dinámicas: pop de nodos

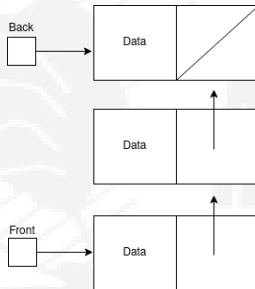


Figure: Representación gráfica de una cola.

# Colas dinámicas: implementación I

```
1 # include <stdio.h>
2 # include <stdlib.h>
3
4 /*Definicion de la estructura*/
5 struct node
6 {
7     int data;
8     struct node *link;
9 };
10
11
12
13
14
15
```

## Colas dinámicas: implementación II

```
16 void main()  
17 {  
18     struct node *front =NULL;  
19     struct node *back  = NULL;  
20     struct node *temp  =NULL;  
21     int n,value,op=1;  
22     printf(" Ingrese un elemento\n");  
23     scanf("%d",&value);  
24     do  
25     { /* Creacion de un nuevo nodo */  
26         temp=(struct node *)malloc(sizeof(struct node));  
27         if(temp==NULL)  
28         { printf("No Memory available\n");  
29           exit(0);  
30         }
```



## Colas dinámicas: implementación III

```
31 temp->data = value;  
32 temp->link=NULL;  
33 if (back == NULL) /*Insercion del primer nodo*/  
34 {  
35     back = temp;  
36     front = back;  
37 }  
38 else /*Insercion del resto de los nodo*/  
39 {  
40     back->link = temp;  
41     back = temp;  
42 }  
43 printf("0 para salir\n");  
44 scanf("%d",&value);  
45 } while (value!=0);
```

## Colas dinámicas: implementación IV

```
46  
47 /*Impresion de toda la FIFO*/  
48 temp=front ;  
49 while (temp!=NULL)  
50 {  
51     printf ("%d" ,temp->data ) ;  
52     temp=temp->link ;  
53 }  
54  
55  
56  
57  
58  
59  
60
```

## Colas dinámicas: implementación V

```
61  do /*Eliminacion de nodos*/
62  {
63      if((front == back) && (back == NULL))
64      {
65          printf(" Vacia" );
66          exit(0);
67      }
68      temp = front;
69      front = (front)->link;
70      if (front == NULL)
71          back = NULL;
72      free(temp);
73      printf("0 para salir\n");
74      scanf("%d",&value);
75  } while (value != 0);
```

## Colas dinámicas: implementación VI

```
76
77  /*Impresion de toda la FIFO*/
78  temp=front ;
79  while (temp!=NULL)
80  {
81      printf ("%d" ,temp->data );
82      temp=temp->link ;
83  }
84
85 }
```

# Colas dinámicas: implementación con punteros I

```
1  # include <stdio.h>
2  # include <stdlib.h>
3
4  /*Definicion de la estructura*/
5  struct node
6  {
7      int data;
8      struct node *link;
9  };
10
11  int menu (void);
12  void push(struct node **,struct node **, int );
13  void pop(struct node **,struct node **);
14  void print(struct node *);
15
```

## Colas dinámicas: implementación con punteros II

```
16 int menu (void)
17 {
18     int op;
19     do
20     {
21         printf("1-Agregar un nodo\n");
22         printf("2-Borrar un nodo\n");
23         printf("3-Imprimir cola\n");
24         printf("4-Salir\n");
25         scanf("%d",&op);
26     } while ((op < 1) || (op > 4));
27     return(op);
28 }
29
30
```

## Colas dinámicas: implementación con punteros III

```
31 void push(struct node **front, struct node **back, int d)
32 {
33     struct node *temp;
34     temp=(struct node *)malloc(sizeof(struct node));
35     if(temp==NULL)
36     {
37         printf("No Memory available\n");
38         exit(0);
39     }
40     temp->data = d;
41     temp->link=NULL;
42     if(*back == NULL) /*Insercion del primer nodo*/
43     {
44         *back = temp;
45         *front = *back;
46     }
```

## Colas dinámicas: implementación con punteros IV

```
46     else /*Insercion del resto de los nodo*/
47     {
48         (*back)->link = temp;
49         *back = temp;
50     }
51 }
52
53 void pop(struct node **front , struct node **back)
54 {
55     struct node *temp;
56     if ((*front == *back) && (*back == NULL))
57     {
58         printf(" Vacia\n" );
59         exit(0);
60     }
```



## Colas dinámicas: implementación con punteros V

```
61  temp = *front ;  
62  *front = (*front)->link ;  
63  if (*back == temp)  
64  {  
65      *back = (*back)->link ;  
66  }  
67  
68  free(temp) ;  
69  }  
70  
71  
72  
73  
74  
75
```

## Colas dinámicas: implementación con punteros VI

```
76 void print(struct node *front)
77 {
78     struct node *temp = NULL;
79     /*Impresion de toda la FIFO*/
80     temp=front;
81     while(temp!=NULL)
82     {
83         printf("%d->", temp->data);
84         temp=temp->link;
85     }
86     printf("\n");
87 }
88
89
90
```

## Colas dinámicas: implementación con punteros VII

```
91 void main()  
92 {  
93     struct node *front  =NULL;  
94     struct node *back   = NULL;  
95     struct node *temp   =NULL;  
96     int dato;  
97     int value;  
98     int op;  
99     do  
100    {   op=menu();  
101        switch(op)  
102        {  
103            case 1:  
104                printf("Ingrese el dato a insertar\n");  
105                scanf("%d",&dato);
```

## Colas dinámicas: implementación con punteros VIII

```
106         push(&front ,&back , dato );
107         break ;
108     case 2:
109         pop(&front ,&back );
110         break ;
111     case 3:
112         print ( front );
113         break ;
114     }
115 } while ( op != 4 );
116
117 }
```

## Colas circulares (ring buffer): introducción I

Una cola circular es esencialmente una cola como las estudiadas anteriormente, con un tamaño o capacidad máxima que continuará volviendo sobre sí misma con un movimiento circular.

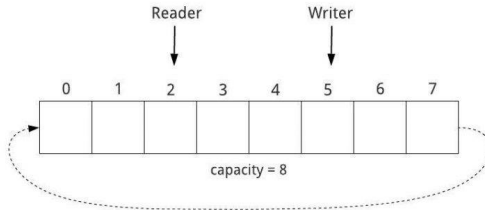


Figure: Representación gráfica de una cola circular

## Colas circulares (ring buffer): introducción II

Ventajas del uso de colas circulares:

- Ofrecen una forma rápida y limpia de almacenar datos FIFO con un tamaño predeterminado
- Como tienen un tamaño fijo, no utilizan memoria dinámica y se evitan fugas de memoria
- Implementación simple
- Las operaciones ocurren en tiempo constante

Desventajas del uso de colas circulares:

- Sólo permiten el almacenamiento de un número fijo de elementos

*¡Muchas gracias!*

Consultas:

[sperez@iua.edu.ar](mailto:sperez@iua.edu.ar)

[drosso@iua.edu.ar](mailto:drosso@iua.edu.ar)