

# Algorítmica y Programación II

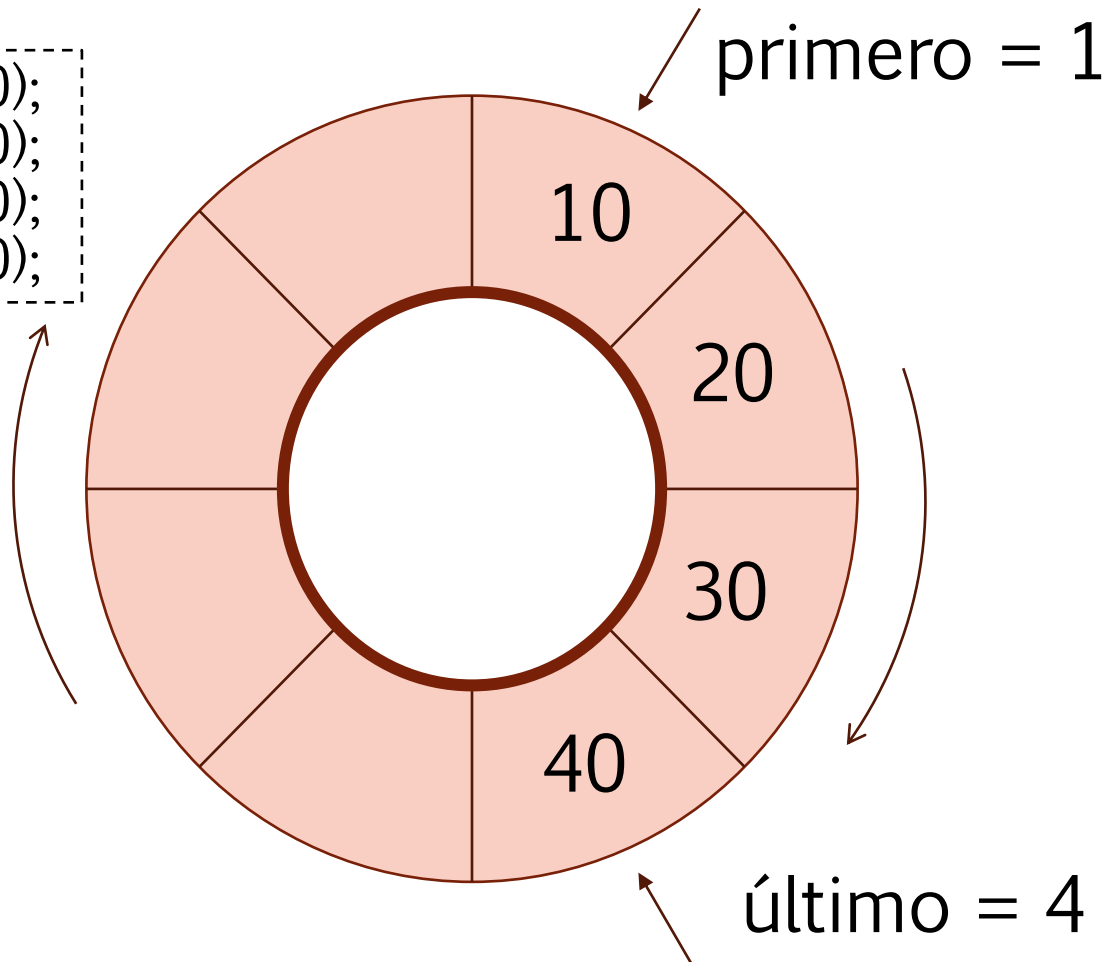
TDA COLA

continuación cola circular y bicola

$\pi$

# COLA circular

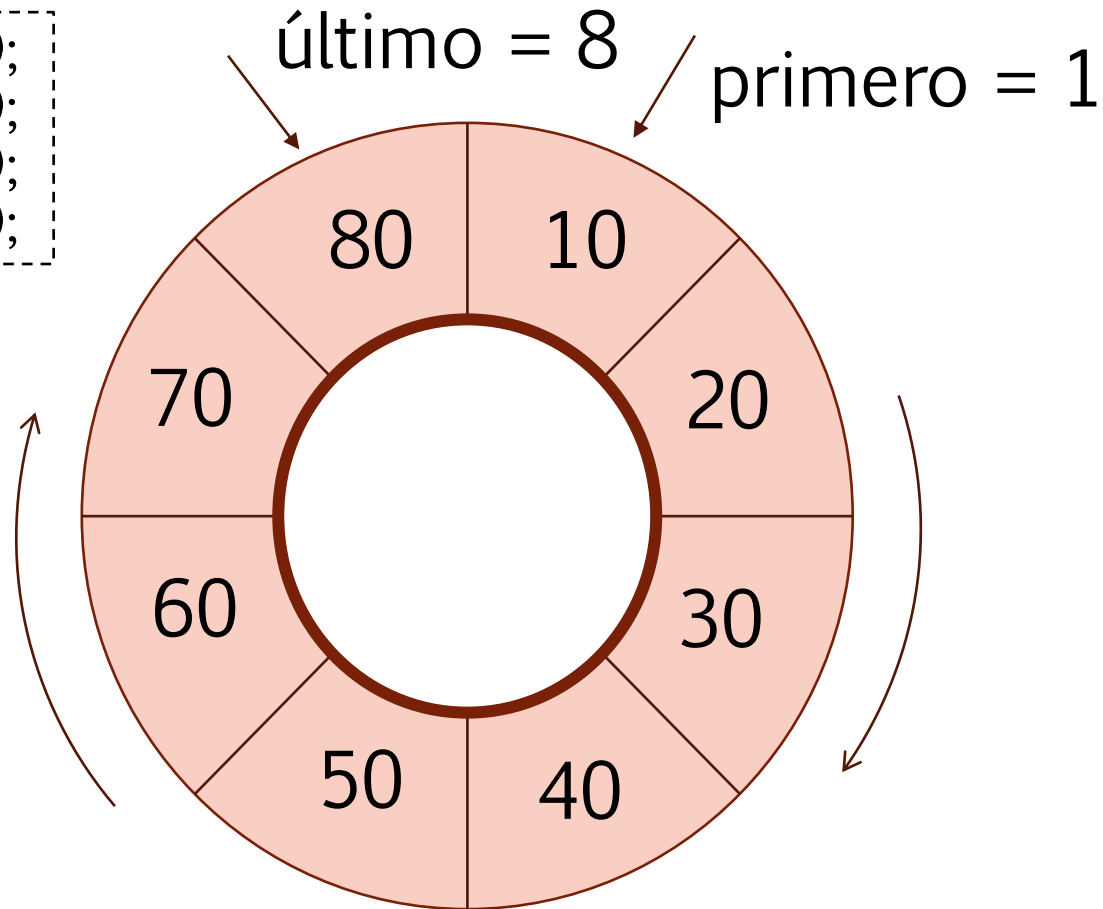
```
Insertar(10);  
Insertar(20);  
Insertar(30);  
Insertar(40);
```



$N=8$

# COLA circular

```
Insertar(50);  
Insertar(60);  
Insertar(70);  
Insertar(80);
```

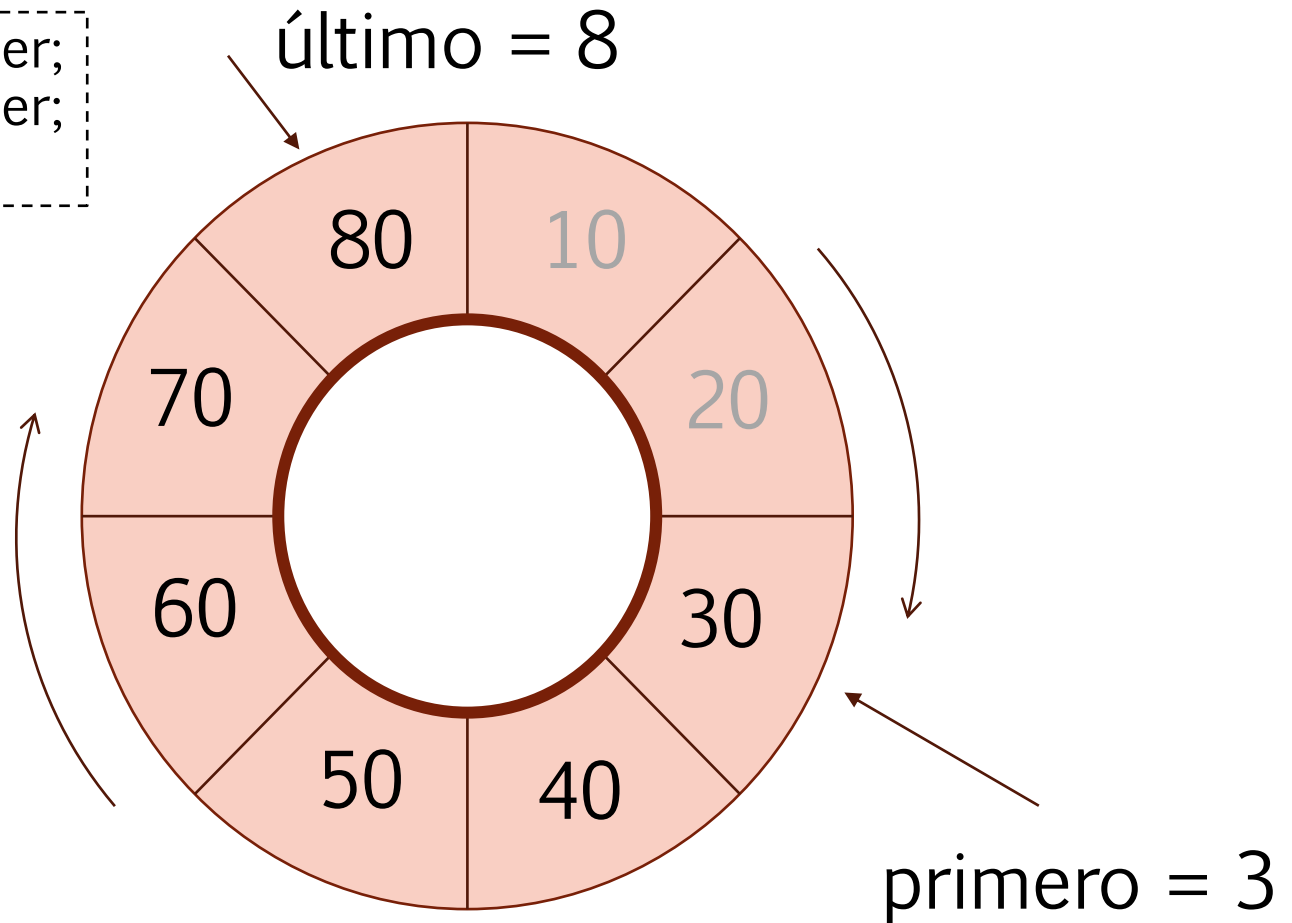


N=8

$\pi$

# COLA circular

10  $\leftarrow$  extraer;  
20  $\leftarrow$  extraer;

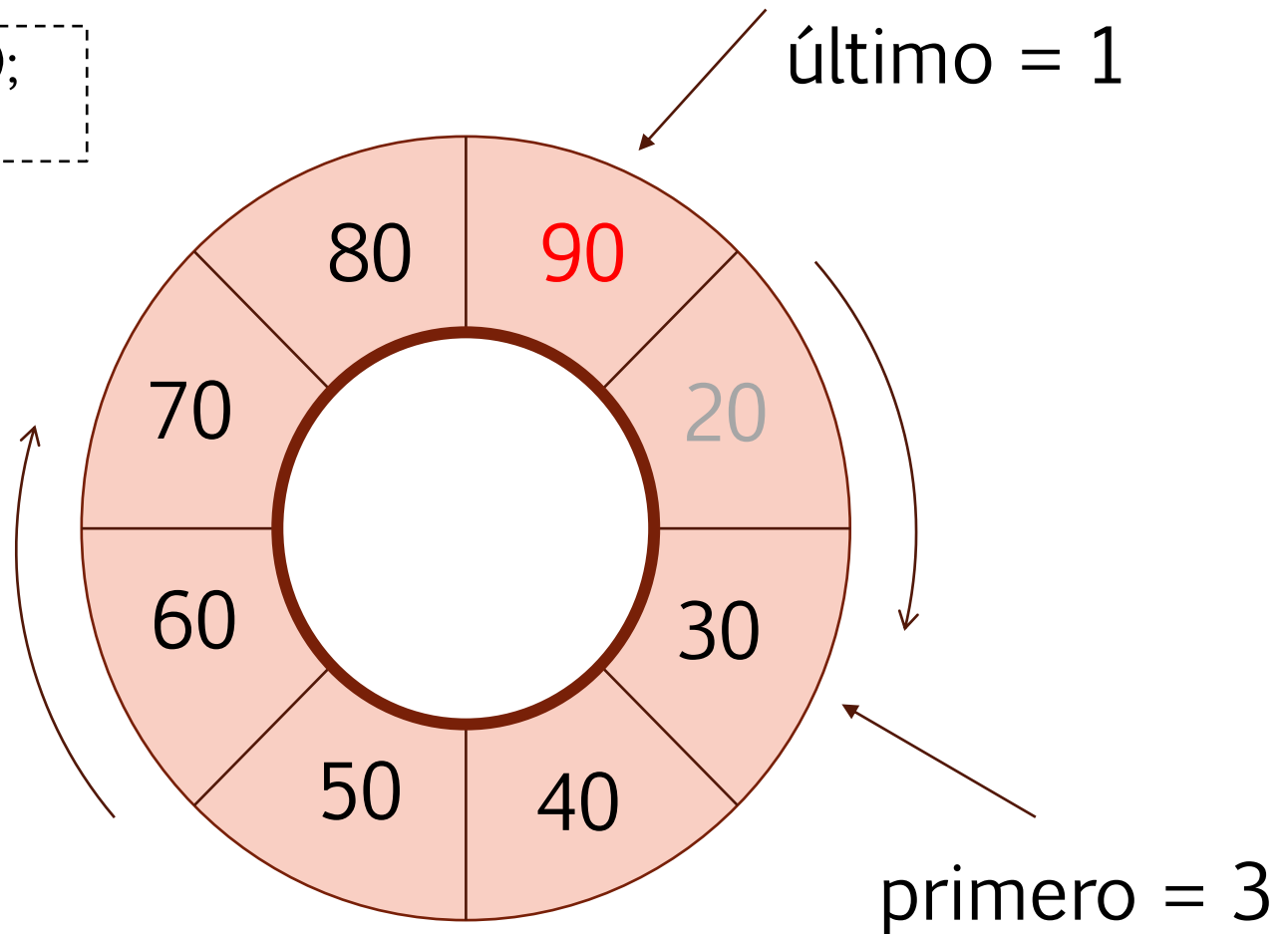


N=8

$\pi$

# COLA circular

Insertar(90);



N=8

## Ejercicio

Implementar una cola circular, **sin utilizar la variable cantidad** de elementos de la cola.

N=8			
ultimo = (ultimo + 1) mod N			primero = (primero + 1) mod N
(-1 + 1)mod N	0		0
(0 + 1)mod N	1		(0 + 1)mod N 1
(1 + 1)mod N	2		(1 + 1)mod N 2
(2 + 1)mod N	3		(2 + 1)mod N 3
(3 + 1)mod N	4		(3 + 1)mod N 4
(4 + 1)mod N	5		(4 + 1)mod N 5
(5 + 1)mod N	6		(5 + 1)mod N 6
(6 + 1)mod N	7		(6 + 1)mod N 7
(7 + 1)mod N	0		(7 + 1)mod N 0

## Implementación - cola circular

```
const
    N=8;
Type
    TElemento = Integer;

    TCola = record
        elementos:array[0..N-1]of TElemento;
        primero,
        ultimo:integer;
    end;
```

```
procedure crear(var cola:TCola);
begin
    with cola do
        elementos:array[0..N-1]of TElemento;
        begin
            primero := -1;
            ultimo := -1;
        end;
    end;

function vacia(const cola:TCola):boolean;
begin
    vacia := cola.primero = -1;
end;

function llena(const cola:TCola):boolean;
begin
    with cola do
        llena := primero = (ultimo + 1) mod N;
    end;
end;
```



## Implementación

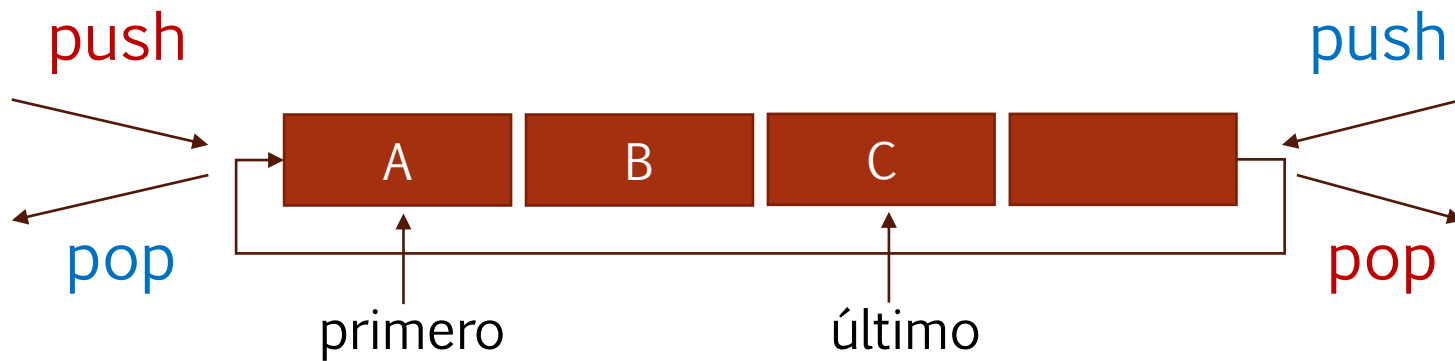
```
procedure meter(var cola:TCola; const E:TElemento);  
begin  
    with cola do  
        begin  
            //ya tengo un elemento para sacar  
            if(primero = -1)then  
                primero := 0;  
            ultimo := (ultimo + 1) mod N;  
            elementos[ultimo] := E;  
        end;  
    end;  
end;
```

## Implementación

```
procedure sacar(var cola:TCola; var E:TElemento);
begin
    with cola do
        begin
            E := elementos[primero];
            if (primero = ultimo)then
                begin
                    primero:= -1;
                    ultimo := -1;
                end
            else
                primero := (primero + 1) mod N;
            end;
        end;
    end;
```

# BICOLA

(doble cola)



## Cola de doble entrada o bicola

- › La bicola se puede considerar como una **variación del TDA cola**.
- › Es un conjunto ordinario de elementos del cual se pueden **añadir o quitar elementos desde cualquier extremo** de la bicola;.
- › **Es una cola bidireccional.**
- › Identificamos sus extremos como izquierdo y derecho (front or back) respectivamente.

## Bicola – operaciones básicas

- › Crear: inicializa la bicola como vacía.
- › Insertar izquierda (push - **insert front**): añade un elemento por el extremo izquierdo.
- › Insertar derecha (**insert back**): añade un elemento por el extremo.
- › Extraer izquierda (**remove front**): extrae el elemento del extremo izquierdo.
- › Extraer derecha (pop - **remove back**): extrae el elemento del extremo derecho.
- › Examina izquierda (peek - **verify front**): retorna el elemento del extremo izquierdo pero no lo remueve.
- › Examina derecha (**verify back**): retorna el elemento del extremo derecho pero no lo remueve.
- › Vacía (**empty**): comprueba si la bicola no tiene elementos.
- › Llena (**full\***): comprueba si la bicola está llena.

## Ejercicios

- a) Implementar el TDA BiCola con las operaciones básicas que permiten su uso.
- b) Implemente las dos variantes de la doble cola:
  - **Doble cola de entrada restringida:** Este tipo de doble cola acepta solamente la inserción de elementos por un extremo; mientras que puede eliminar por ambos.
  - **Doble cola de salida restringida:** Este tipo de doble cola acepta solamente la eliminación de elementos por un extremo; mientras que puede insertar por ambos.

## TDA - Bibliografía

- Data Structures. Nalle Dale.
- Algoritmos, Datos y programas. Armando, De Gusti.
- Estructuras de datos y algoritmos. Mark, Weiss.
- [https://en.wikipedia.org/wiki/Queue\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Queue_(abstract_data_type))
- [https://en.wikipedia.org/wiki/Double-ended\\_queue](https://en.wikipedia.org/wiki/Double-ended_queue)