

# Colas - Operaciones

Operación	Descripción
public Cola ()	Crea una cola vacía.
public void <b>encolar</b> (Tipo elemento)	Inserta un elemento al final de la cola ( <i>enqueue</i> ).
public Tipo <b>desencolar</b> ()	Consulta y saca el primer elemento ( <i>dequeue</i> ), sólo si la cola no está vacía.
public Tipo <b>primero</b> ()	Consulta el primer elemento, sólo si la cola no está vacía.
public boolean <b>esVacia</b> ()	Indica si la cola está vacía.
public int <b>talla</b> ()	Consulta el nº de elementos.

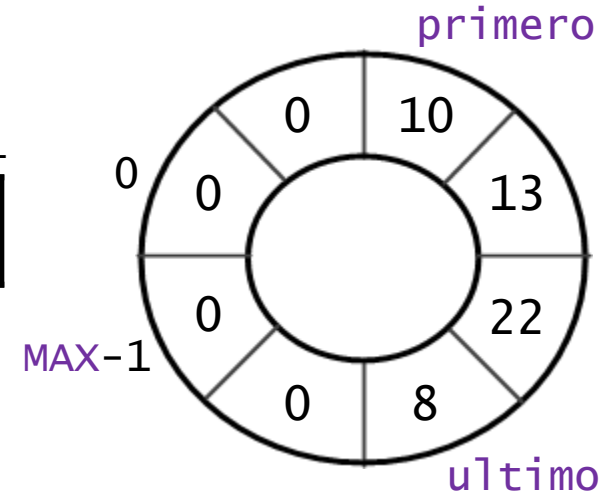
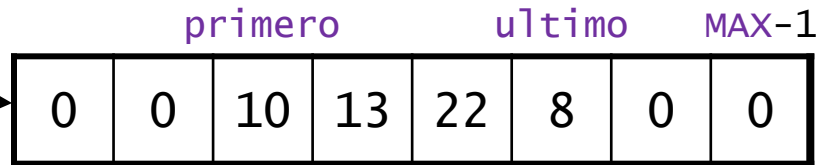
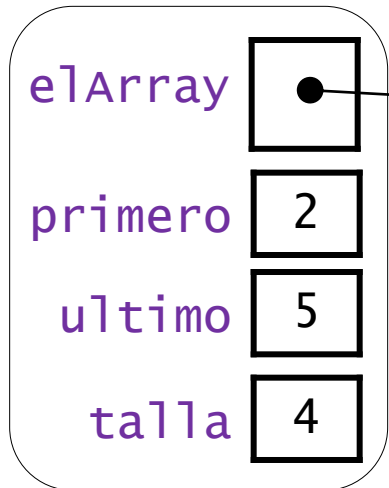
Se puede conseguir implementar estas operaciones con coste constante.

# Colas – Implementación con arrays

- Una cola se puede implementar utilizando un array (**elArray**) que almacene los datos, dos índices (**primero** y **ultimo**) que marquen los dos puntos de acceso a la cola y una constante que define la dimensión máxima del array (**MAX**).
- Para poder reutilizar todas las posiciones del array sin tener que desplazar elementos y así llegar a su final con todas sus posiciones ocupadas, se considera el array como una *estructura circular*, sin principio ni fin, donde después del último elemento va el primero.
- Además, y para poder distinguir las situaciones de cola sin elementos y cola con el número de elementos máximo (**MAX**), se añade un atributo (**talla**) que es el número de elementos de la cola.

# Colas – Implementación con arrays

- Cola cuyos datos son de tipo `int`:



```
public class ColaIntArray {  
    private int[] elArray;  
    private int primero, ultimo, talla;  
    private static final int MAX = ...;  
  
    // Implementación de las operaciones:  
    ...  
}
```

# Colas – Implementación con arrays

- Operación constructora `ColaIntArray`: Crea el array e inicializa los puntos de acceso, indicando que la cola está vacía.

```
public ColaIntArray () {  
    elArray = new int[MAX];  
    talla = 0; primero = 0; ultimo = -1;  
}
```

- Operación `encolar`:

```
public void encolar (int x) {  
    if (talla==MAX)  
        System.out.println("No caben más elementos");  
    else {  
        ultimo = (ultimo+1)%MAX;  
        elArray[ultimo] = x;  
        talla++;  
    }  
}
```

# Colas – Implementación con arrays

- Operaciones **desencolar** y **primero**: Sólo aplicables sobre una cola no vacía.

```
public int desencolar () {  
    int x = elArray[primero];  
    primero = (primero+1)%MAX;  
    talla--;  
    return x;  
}
```

```
public int primero () {  
    return elArray[primero];  
}
```

# Colas – Implementación con arrays

- Operación **esVacia**:

```
public boolean esVacia () {  
    return talla==0;  
}
```

- Operación **talla**:

```
public int talla () {  
    return talla;  
}
```