



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela

## Listas Enlazadas

1. Estructuras de datos
2. Definición de la clase ListaEnlazada
3. Operación agregar:
  - a. Al frente
  - b. Al final
  - c. Al medio

Linked List data structures be like:



I know a guy who knows a guy

# Estructuras de datos

---

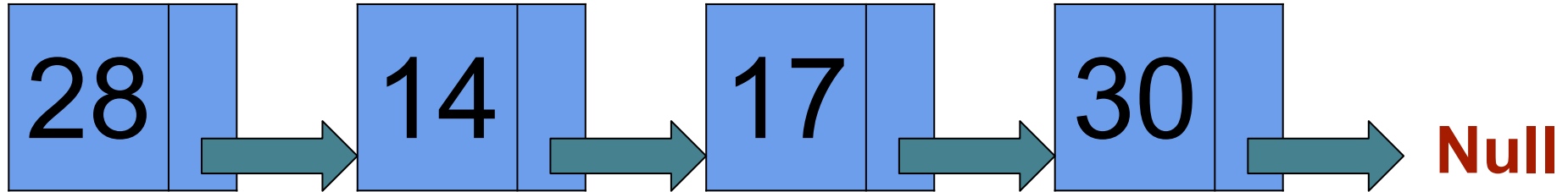
1. Una **estructura de datos** es una forma específica de organizar datos en una computadora.
2. Ejemplos:
  - ◆ arreglos, listas, sets, árboles, etc.

# Gestión de datos con Arreglos

---

1. Se debe conocer el tamaño del arreglo anticipadamente
  - ◆ Muy pequeño : no se puede agregar elementos
  - ◆ Muy grande : se pierde espacio
2. No es posible insertar ( o eliminar ) un elemento delante de otros de manera sencilla
  - ◆ Se debe mover todos los elementos, y se debe asumir que se cuenta con suficiente espacio
3. Algunas operaciones pueden ser muy lentas

nums[0]	nums[1]	nums[2]	nums[3]
28	14	17	30



# Definición:

## Clase ListaEnlazada

# Definición de una clase Lista Enlazada

---

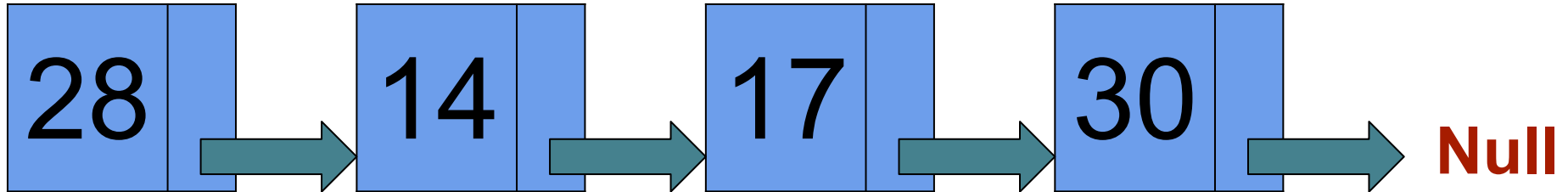
```
public class ListaEnlazada {  
  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
  
    . . .  
}
```

```
class Nodo {  
    int valor;  
    Nodo siguiente = null;  
    Nodo ( int valor ) {  
        this.valor = valor;  
    }  
}
```

# Definición de una clase Lista Enlazada

```
public class ListaEnlazada {  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
    . . .  
}
```

```
class Nodo {  
    int valor;  
    Nodo siguiente = null;  
    Nodo ( int valor ) {  
        this.valor = valor;  
    }  
}
```





# Definición de una clase Lista Enlazada

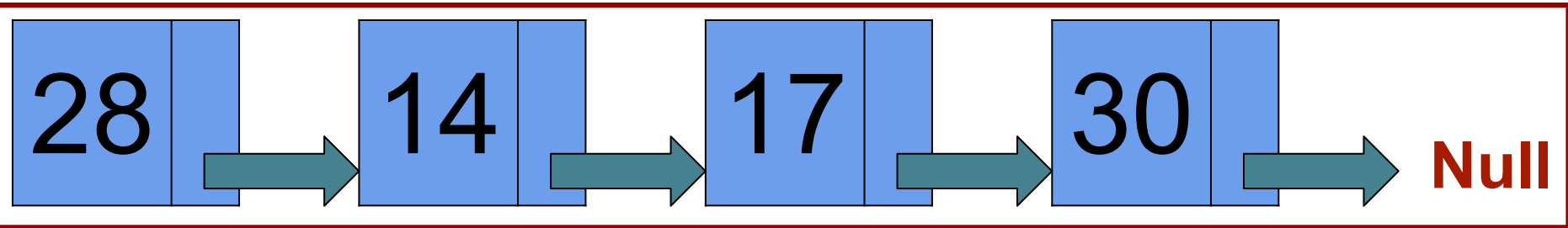
```
public class ListaEnlazada {
```

```
protected Nodo cabeza = null;
```

```
protected Nodo cola = null;
```

```
...
```

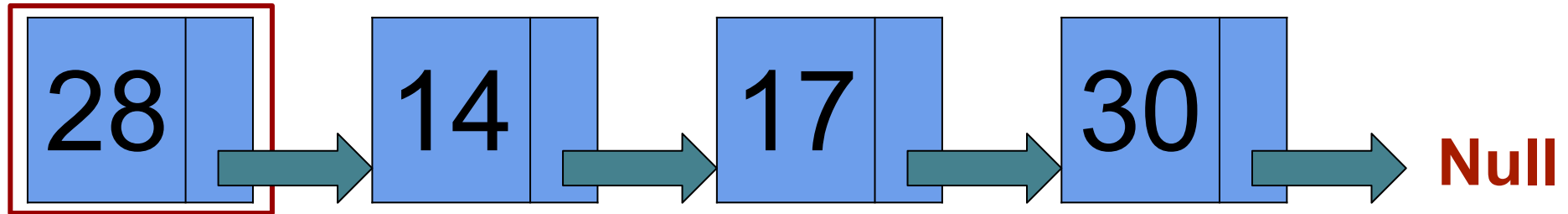
```
class Nodo {  
    int valor;  
    Nodo siguiente = null;  
    Nodo ( int valor ) {  
        this.valor = valor;  
    }  
}
```



# Definición de una clase Lista Enlazada

```
public class ListaEnlazada {  
  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
  
    . . .  
}
```

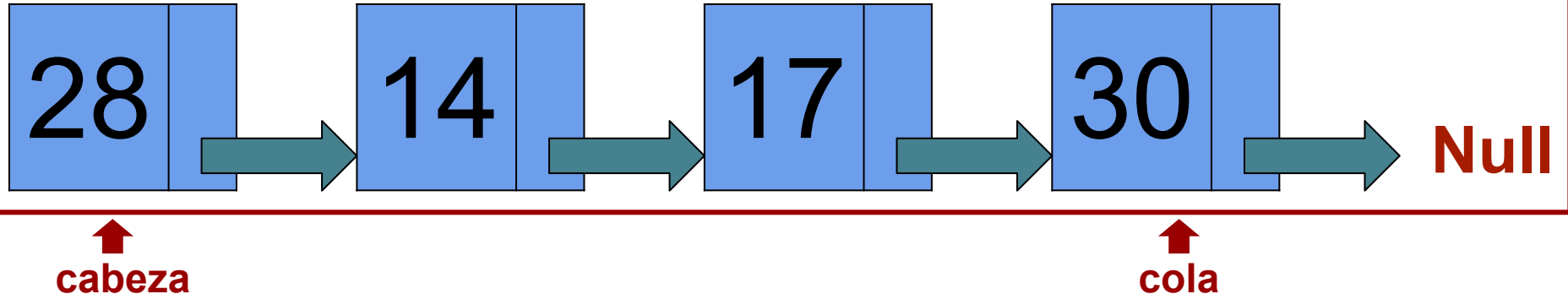
```
class Nodo {  
    int valor;  
    Nodo siguiente = null;  
    Nodo ( int valor ) {  
        this.valor = valor;  
    }  
}
```



# Definición de una clase Lista Enlazada

```
public class ListaEnlazada {  
  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
  
    . . .  
}
```

```
class Nodo {  
    int valor;  
    Nodo siguiente = null;  
    Nodo ( int valor ) {  
        this.valor = valor;  
    }  
}
```

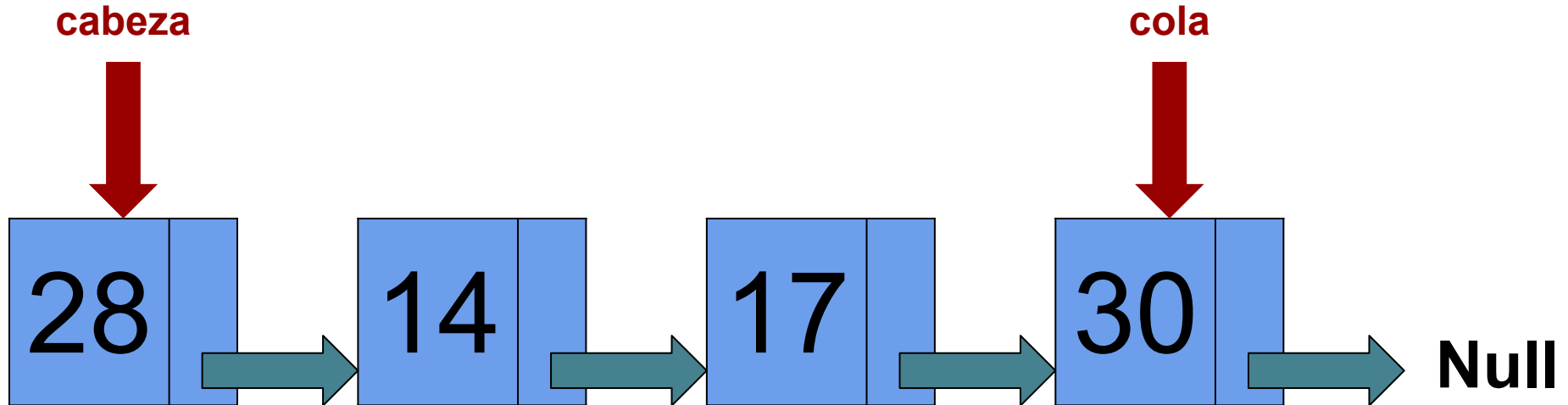


# Operación #1:

**Agregar un “valor” al frente de una Lista Enlazada**

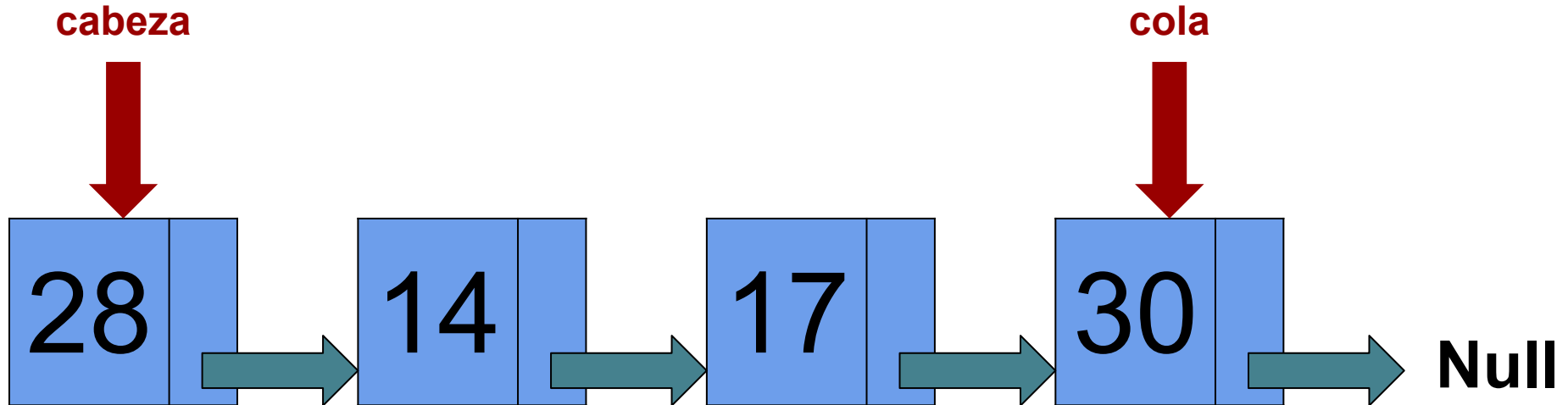
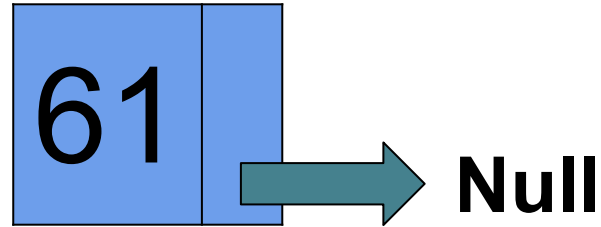
# Agregar un valor al frente

---

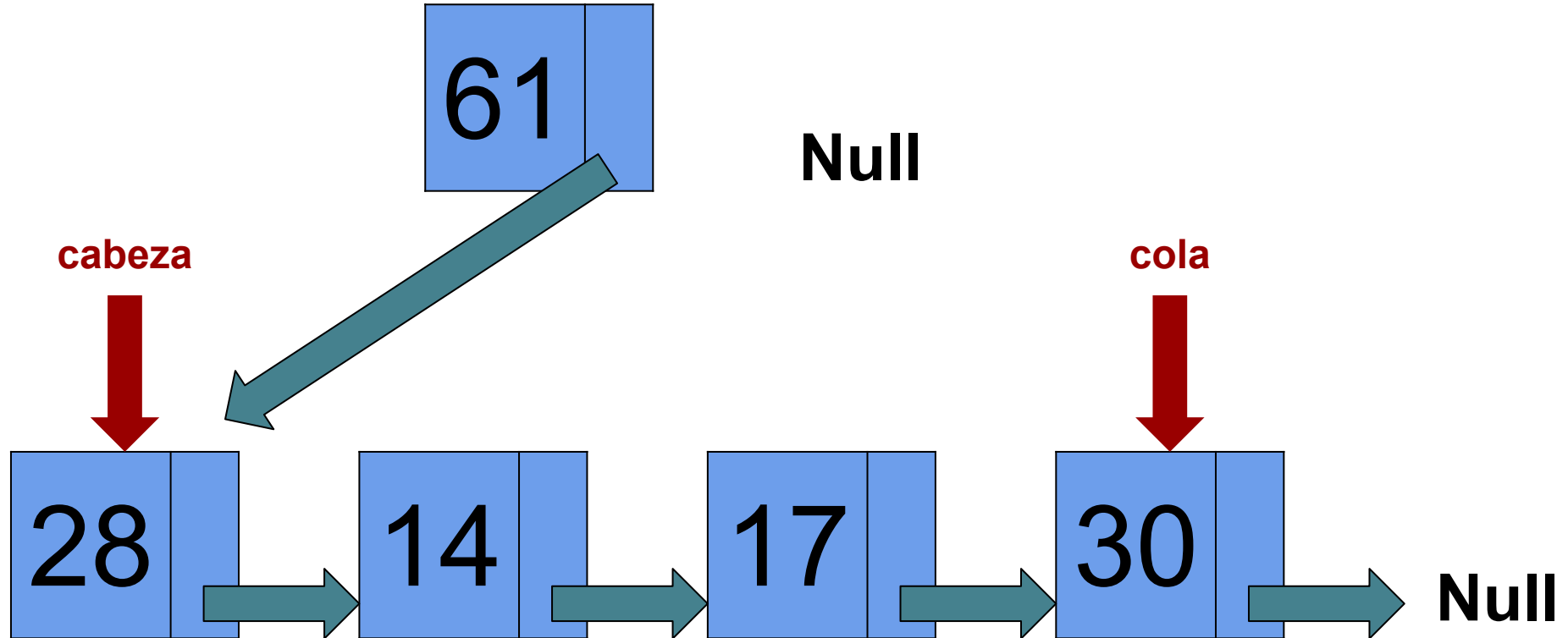


# Agregar un valor al frente

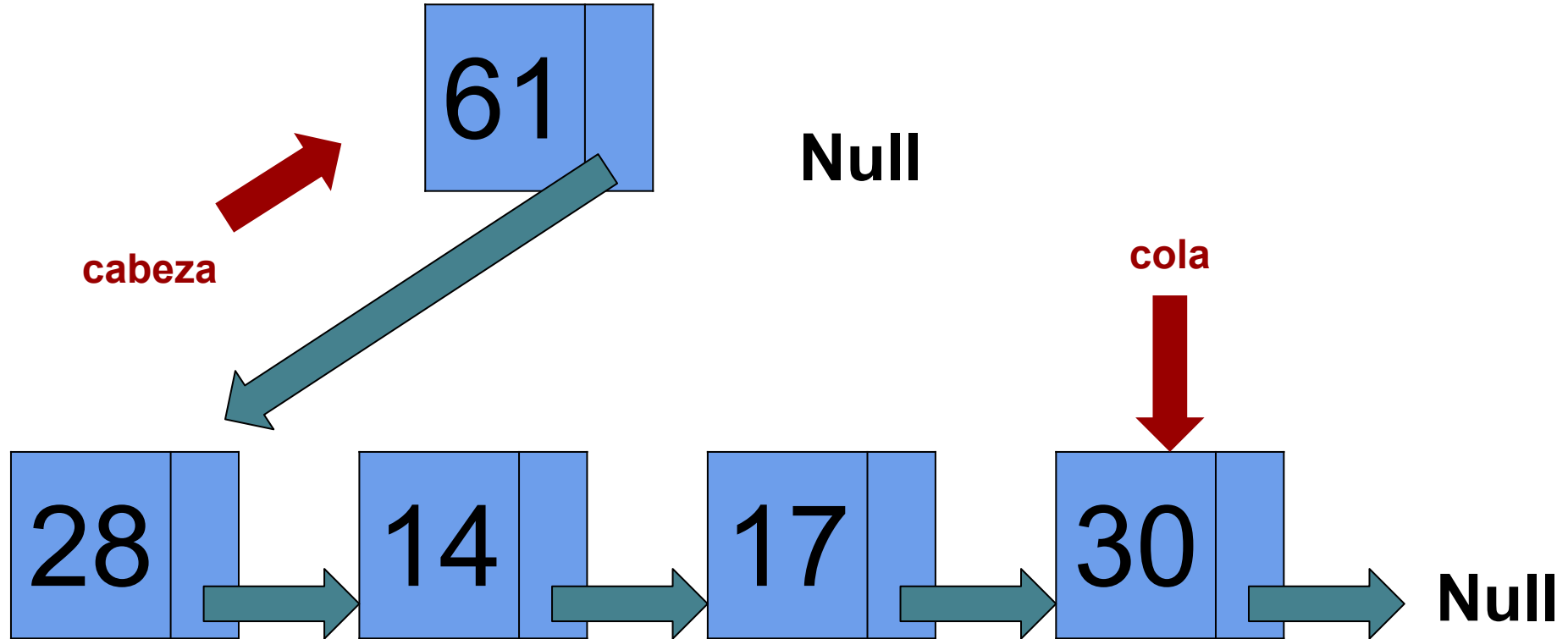
---



# Agregar un valor al frente



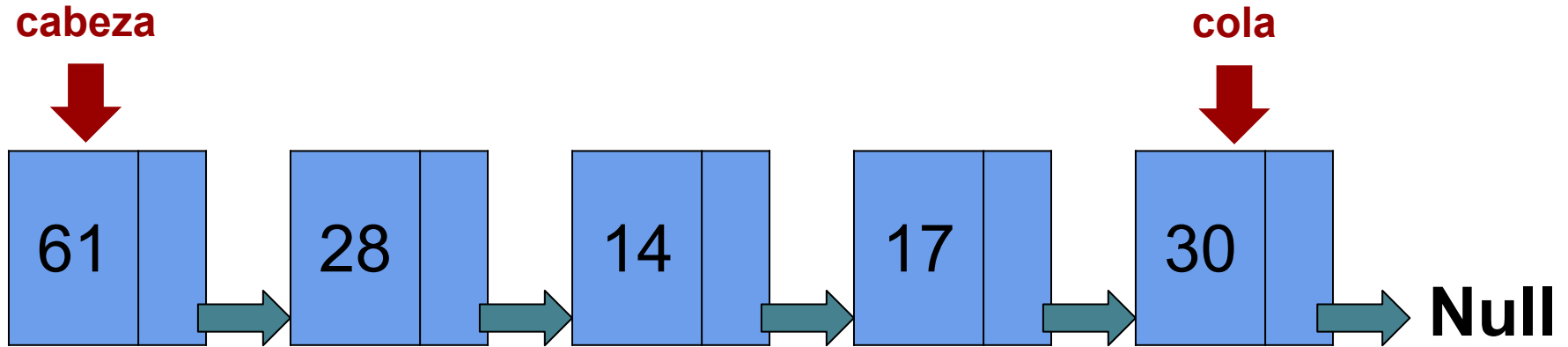
# Agregar un valor al frente





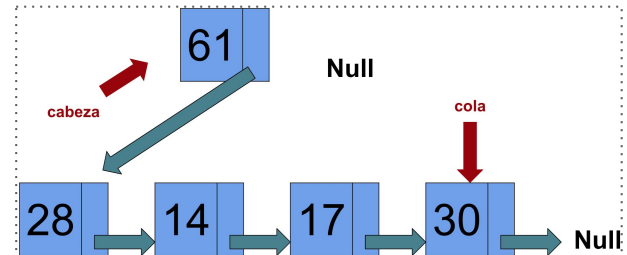
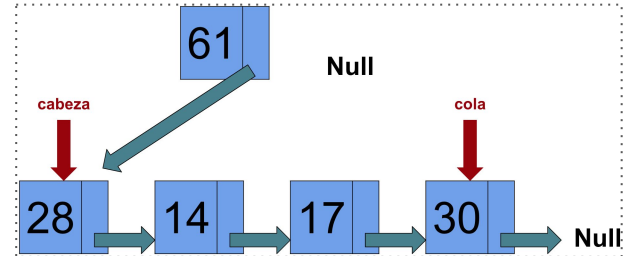
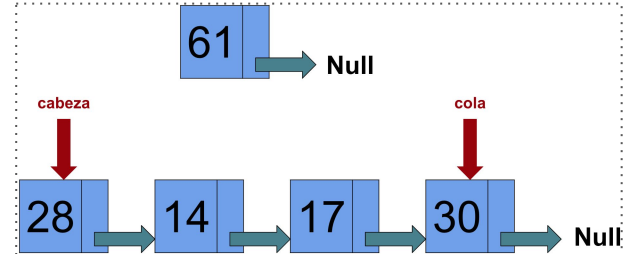
# Agregar un valor al frente

---



# Agregar un valor al frente

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
  
    public void agregarFrente ( int valor ) {  
        // crear nuevo nodo  
        Nodo nuevoNodo = new Nodo(valor);  
        // asignar siguiente al nuevo nodo  
        nuevoNodo.siguiente = cabeza;  
        // definir nuevoNodo como cabeza  
        cabeza = nuevoNodo;  
    }  
}
```



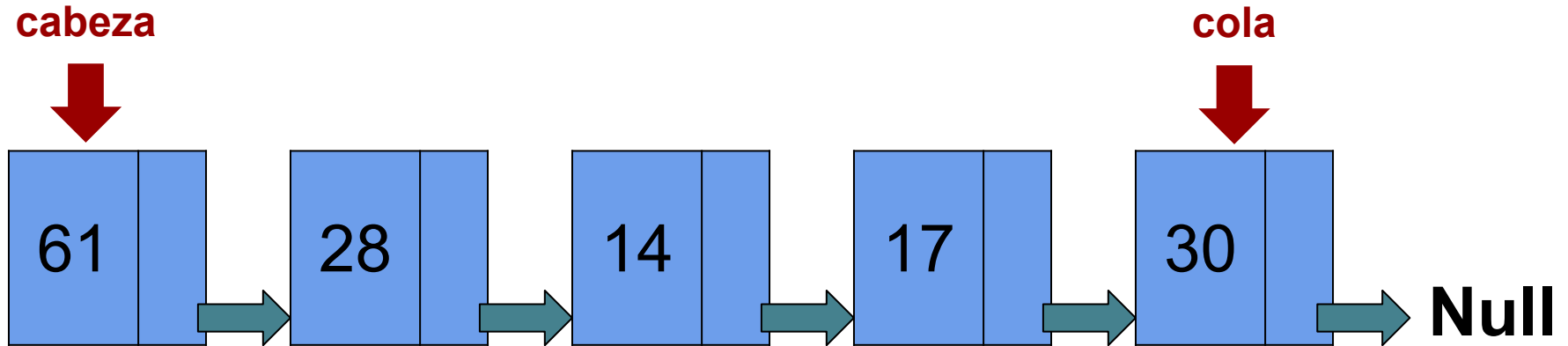
# Agregar un valor al comienzo

---

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cabeza = null;  
    protected Nodo cola = null;  
  
    public void agregarFrente ( int valor ) {  
        Nodo nuevoNodo = new Nodo(valor);  
        nuevoNodo.siguiente = cabeza;  
        cabeza = nuevoNodo;  
        // Lista vacía  
        if ( nuevoNodo.siguiente == null ) {  
            cola = nuevoNodo;  
        }  
    }  
}
```

# Agregar un valor al comienzo

---

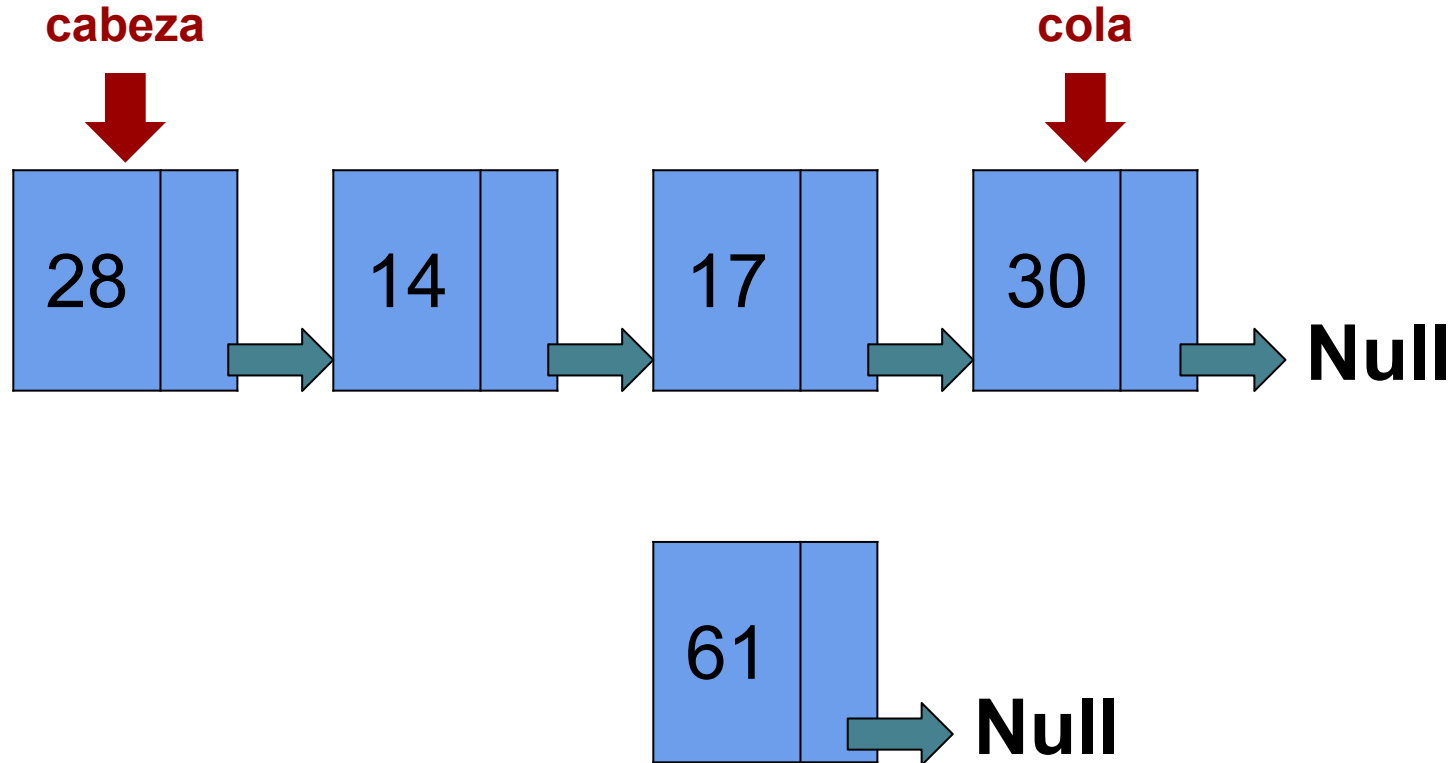


# Operación #2:

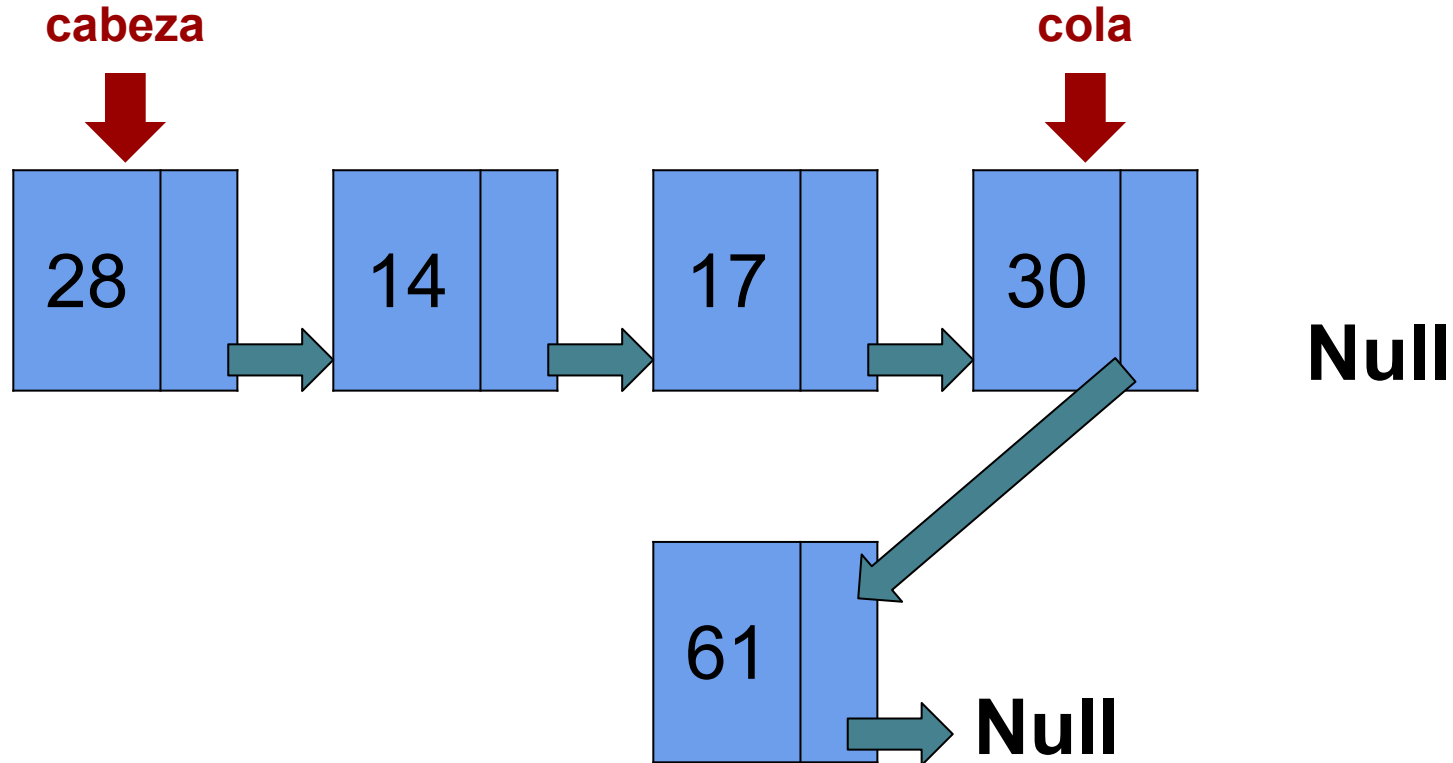
**Agregar un “valor” a la cola de la lista enlazada**

# Agregar un valor a la cola

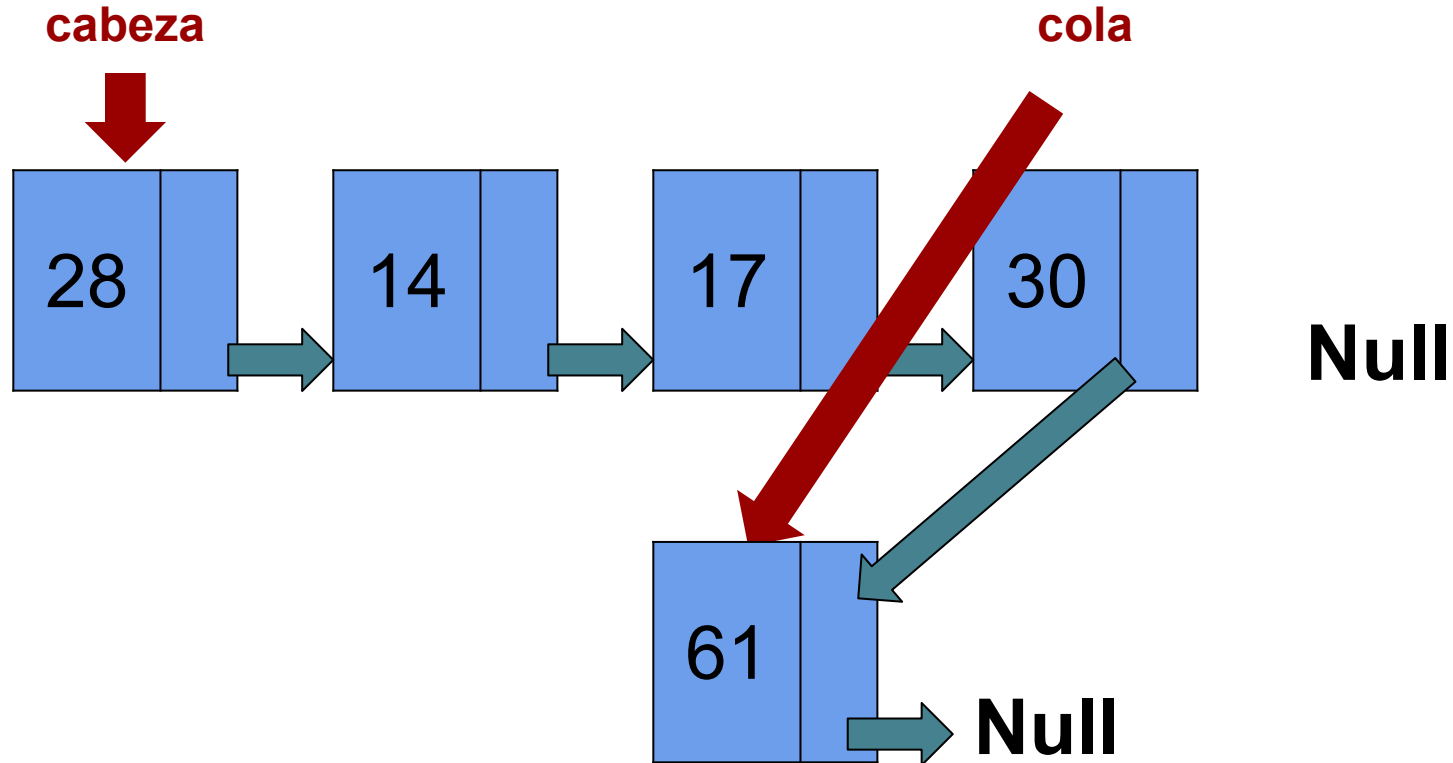
---



# Agregar un valor a la cola



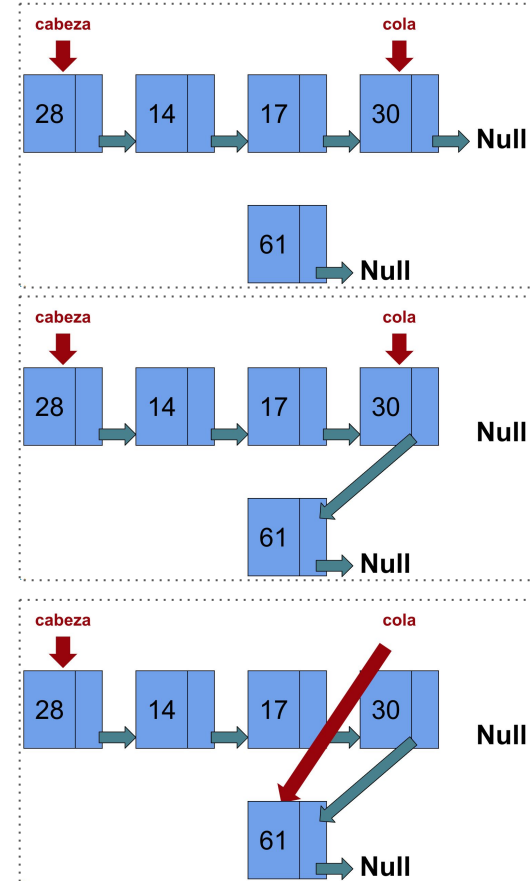
# Agregar un valor a la cola





# Agregar un valor a la cola

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cola = null;  
  
    public void agregarFinal ( int value ) {  
        // crear nuevo nodo  
        Nodo nuevoNodo = new Nodo(valor);  
        // asignar siguiente al nodo cola  
        cola.siguiente = nuevoNodo;  
        // asignar nuevo nodo como cola  
        cola = nuevoNodo;  
    }  
}
```



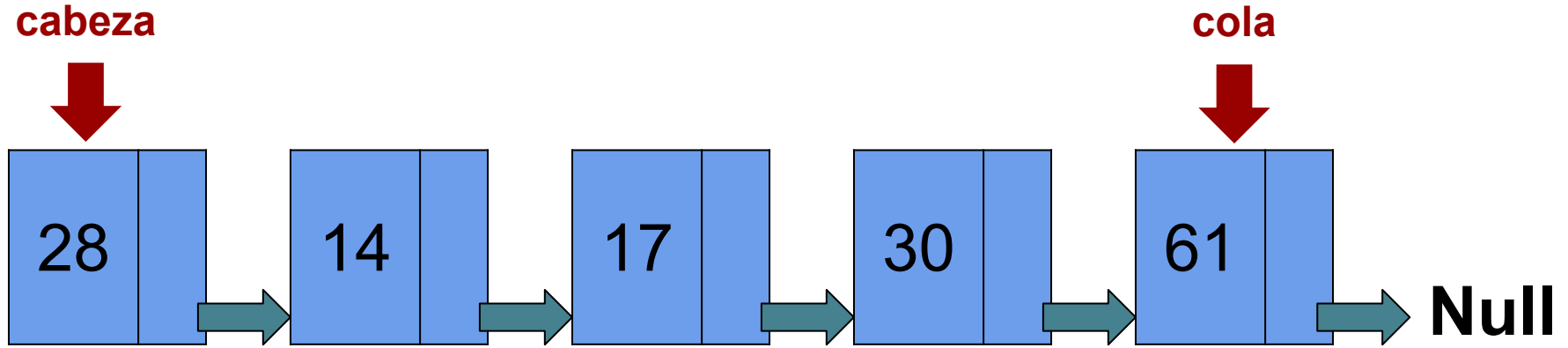
# Agregar un valor a la cola

---

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cola = null;  
  
    public void agregarFinal ( int value ) {  
        Nodo nuevoNodo = new Nodo(valor);  
        // Lista vacía  
        if ( cola == null ){  
            cabeza = nuevoNodo;  
        } else {  
            cola.siguiente = nuevoNodo;  
        }  
        cola = nuevoNodo;  
    }  
}
```

# Agregar un valor a la cola

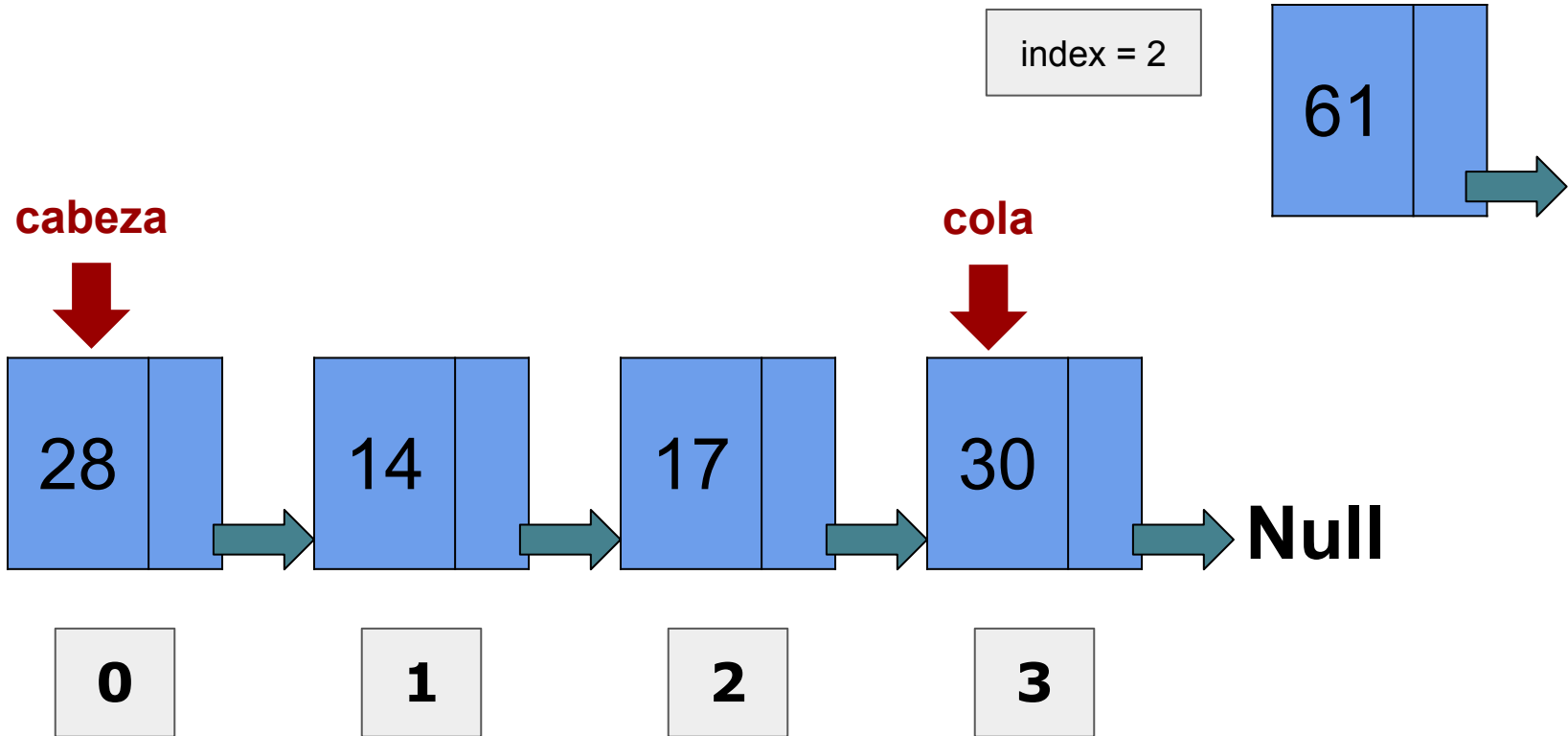
---



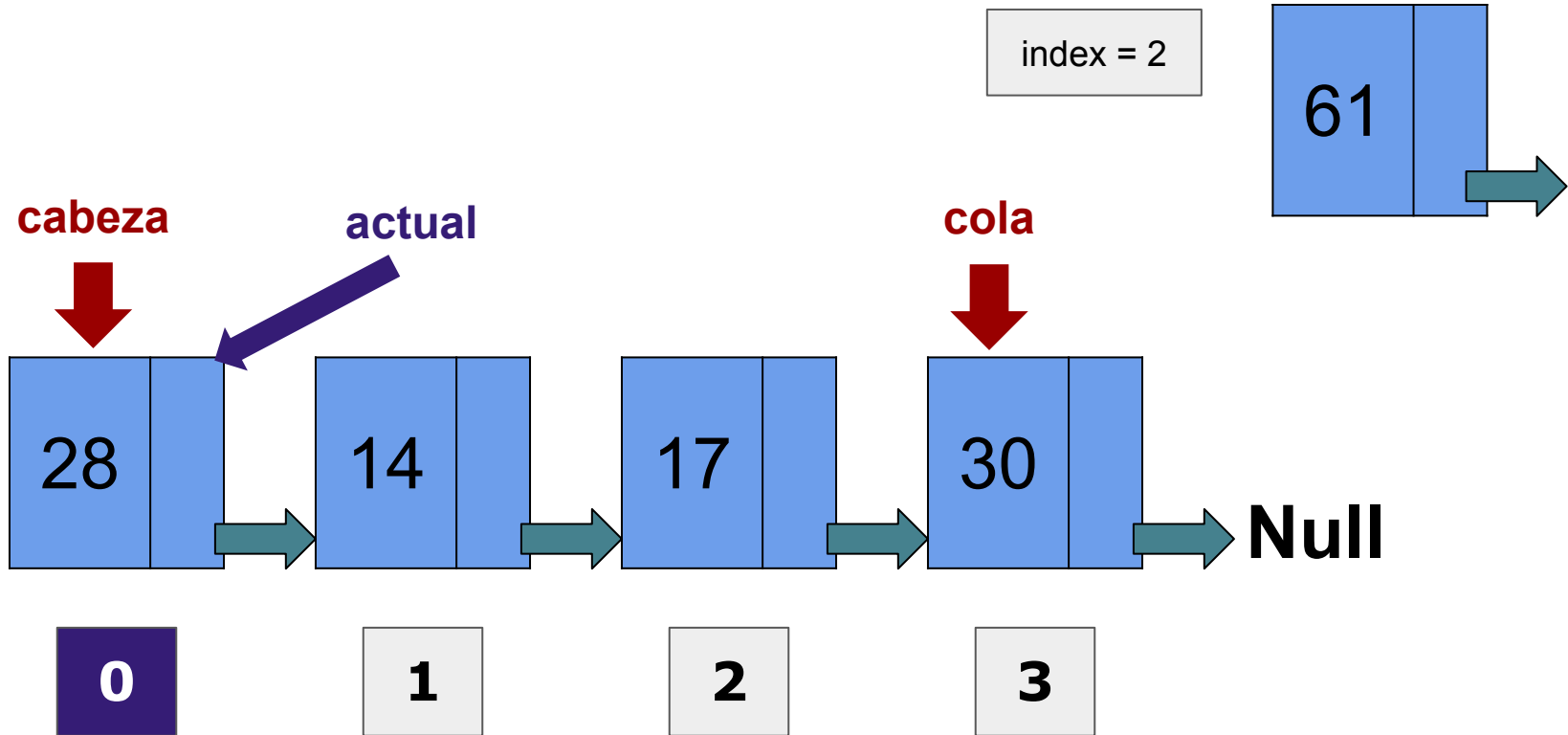
# **Operación #3:**

**Agregar un valor al medio de la lista enlazada (en posición ( índice ) específica )**

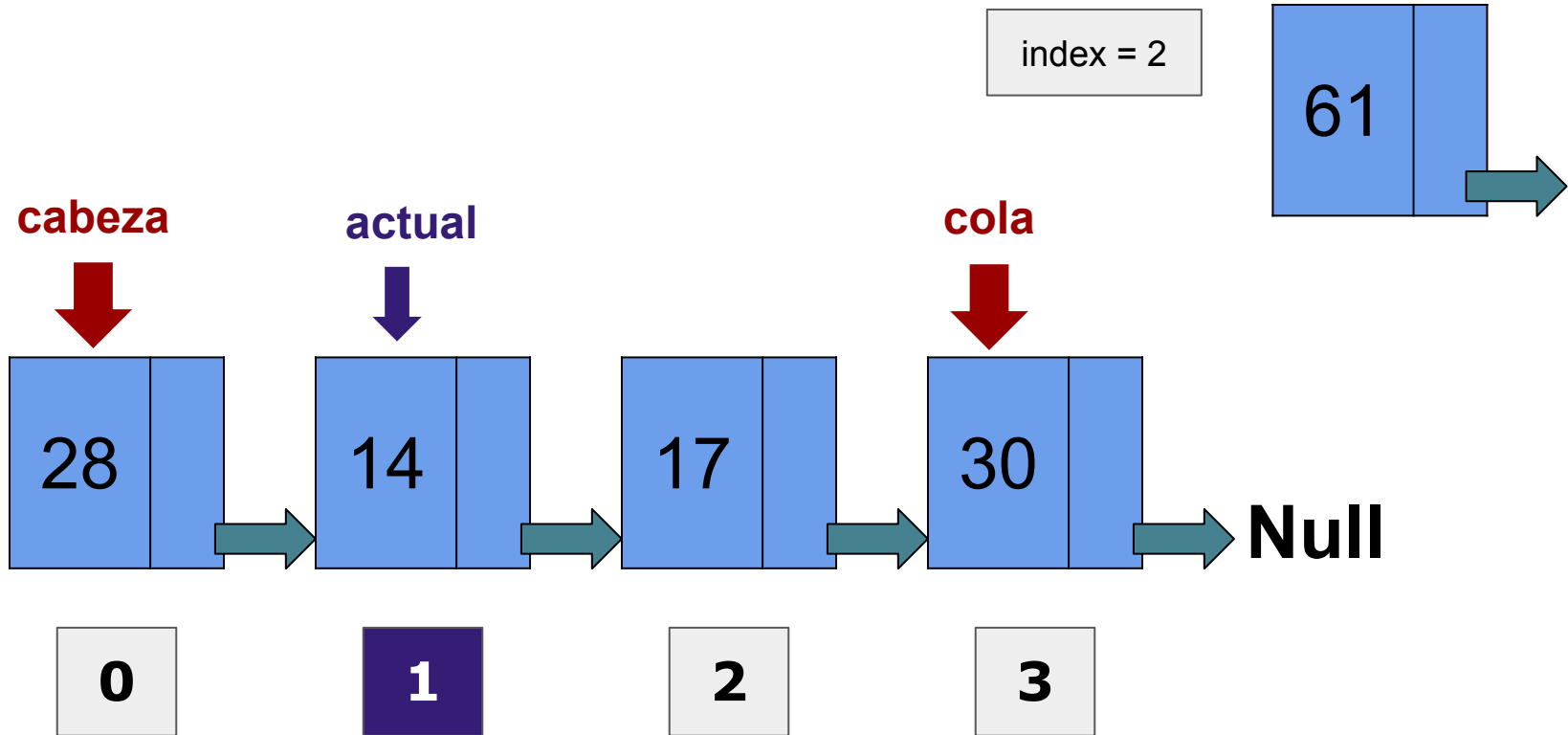
# Agregar en posición específica



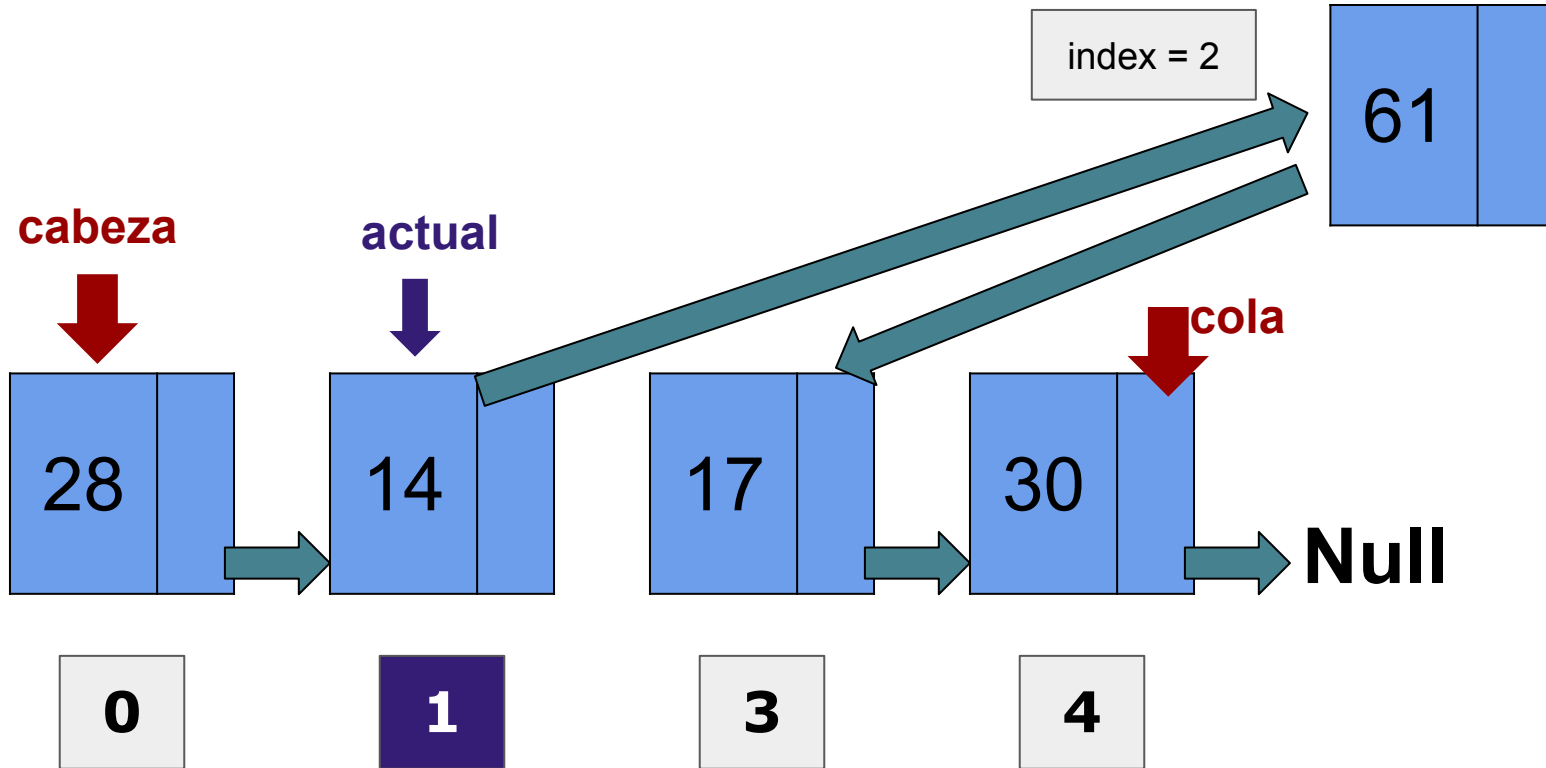
# Agregar en posición específica



# Agregar en posición específica



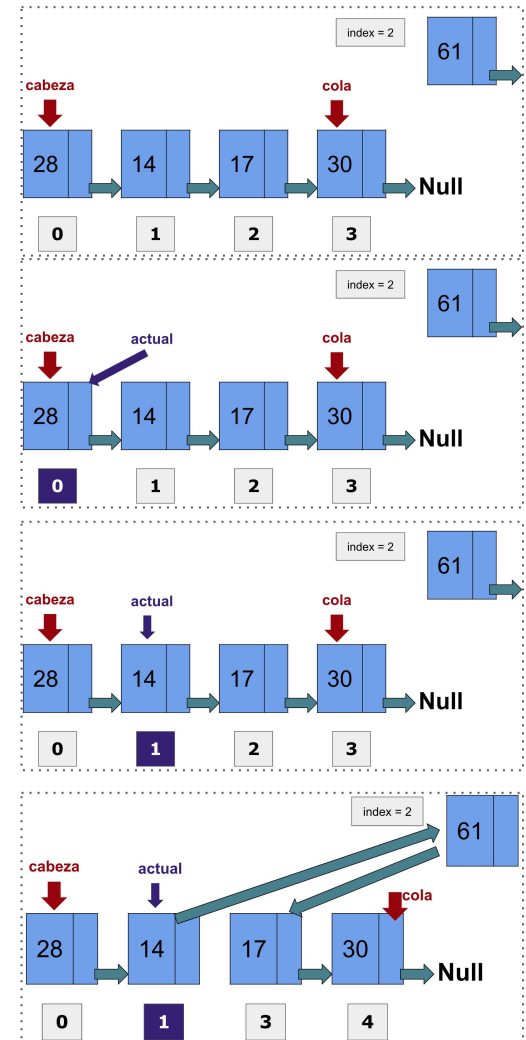
# Agregar en posición específica





# Agregar en posición específica

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cola = null;  
  
    public void agregarEnIndice ( int index, int valor ) {  
        // crear nuevo nodo  
        Nodo nuevoNodo = new Nodo(valor);  
        // definir primer nodo actual  
        Nodo actual = cabeza;  
        // definir nodo actual por índice  
        for ( int i = 0; i < index - 1; i++ ){  
            actual = actual.siguiente;  
        }  
        nuevoNodo.siguiente = actual.siguiente;  
        actual.siguiente = nuevoNodo;  
    }  
}
```

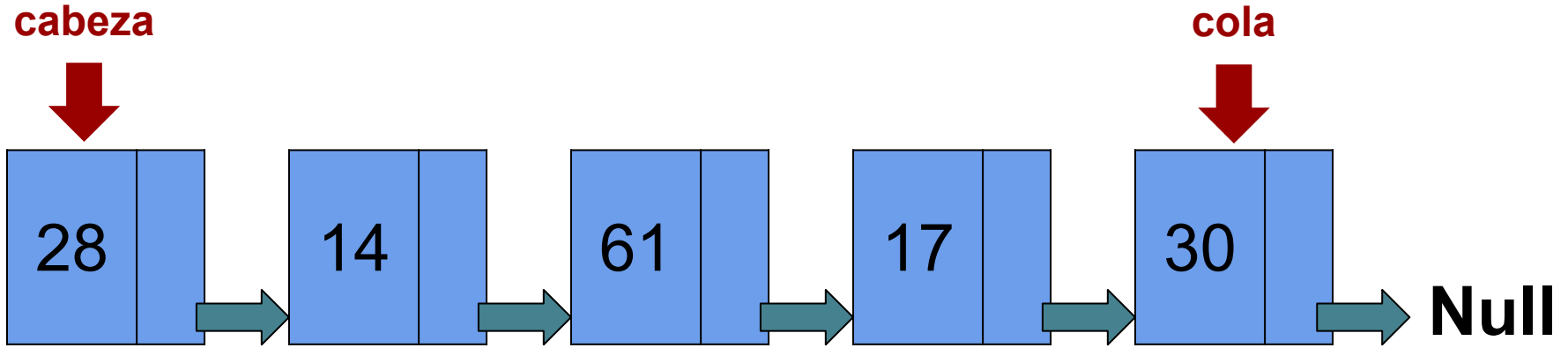


```
public void agregarEnIndice ( int index, int value ) {  
    if (index < 0) {  
        throw new IndexOutOfBoundsException();  
    } else if ( index == 0) { // agregar en la cabeza  
        agregarFrente (valor);  
    } else {  
        Nodo nuevoNodo = new Nodo(valor);  
        Nodo actual = cabeza;  
        for ( int i = 0; i < index - 1; i++ ){  
            if ( actual == null ) {  
                throw new IndexOutOfBoundsException();  
            }  
            actual = actual.siguiente;  
        }  
        if ( actual.siguiente == null ) { //añadir al final  
            cola = nuevoNodo;  
        } else {  
            nuevoNodo.siguiente = actual.siguiente;  
            actual.siguiente = nuevoNodo;  
        }  
    }  
}
```



# Agregar en posición específica

---



# **Comparación entre:**

## **Listas Enlazadas y Arreglos**

# Listas Enlazadas v/s Arreglos

---

1. No se debe conocer el tamaño de la lista que se crea anticipadamente
2. Es posible insertar elementos fácilmente en la posición que se requiera



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela