



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela

## Listas Enlazadas

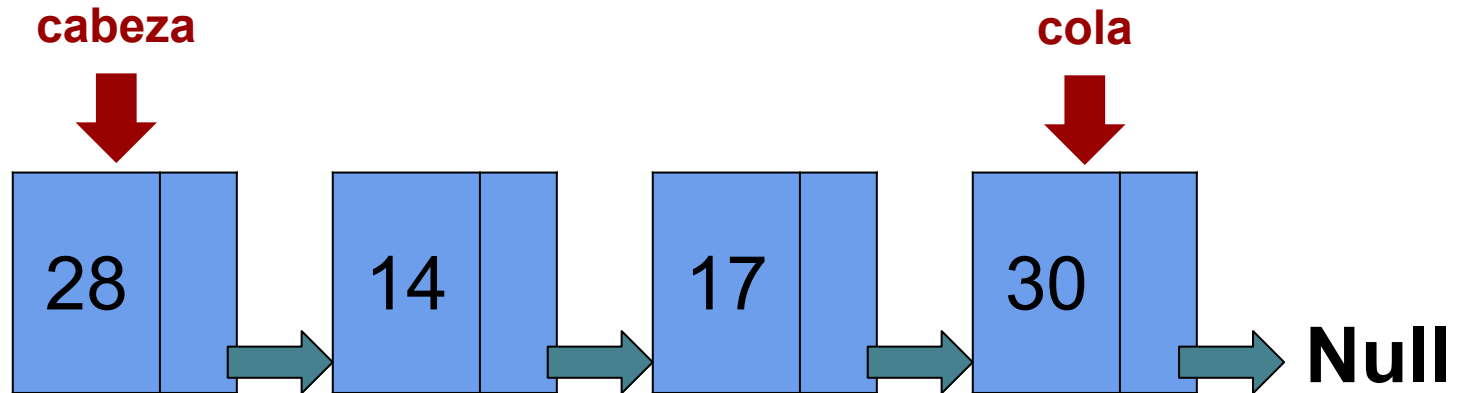
1. Operaciones:
  - a. remover desde el frente
  - b. remover según índice
  - c. remover desde la cola

# Operación 1:

**Remover valores desde el frente en una Lista Enlazada**

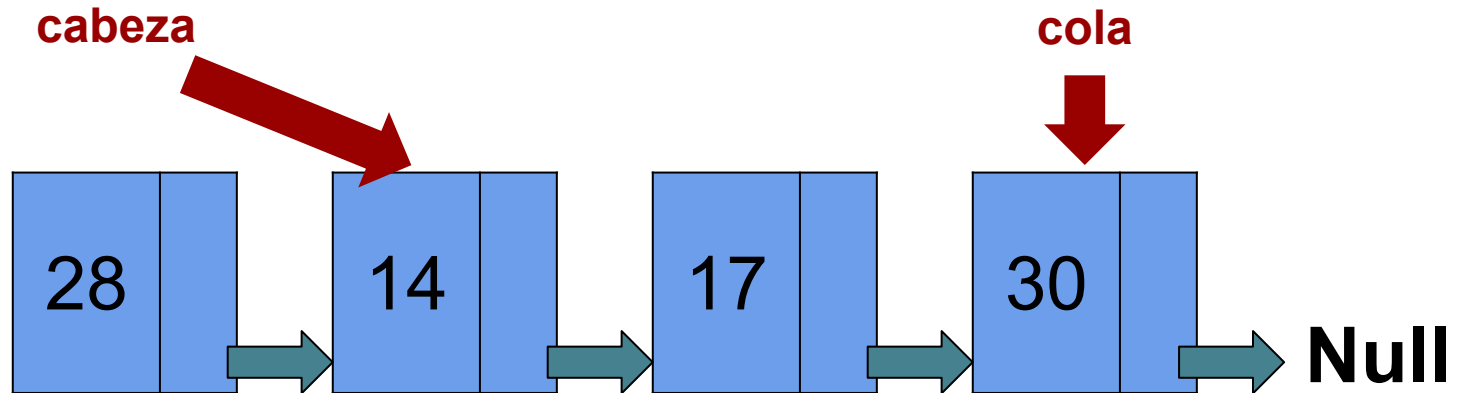
# Lista enlazada: remover desde el frente

---



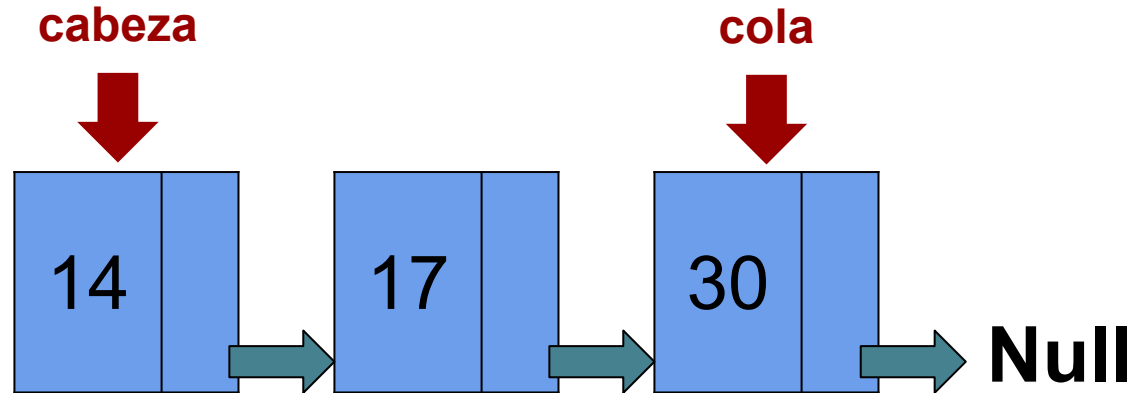
# Lista enlazada: remover desde el frente

---



# Lista enlazada: remover desde el frente

---



# Lista enlazada: remover desde el frente

---

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Node cabeza = null;  
  
    public void removerDesdeFrente () {  
        cabeza = cabeza.siguiente;  
    }  
  
}
```

# Lista enlazada: remover desde el frente

---

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Node cabeza = null;  
  
    public void removerDesdeFrente () {  
        if ( head != null ) {  
            cabeza = cabeza.siguiente;  
        }  
        if ( head == null ) {  
            cola = null;  
        }  
    }  
}
```

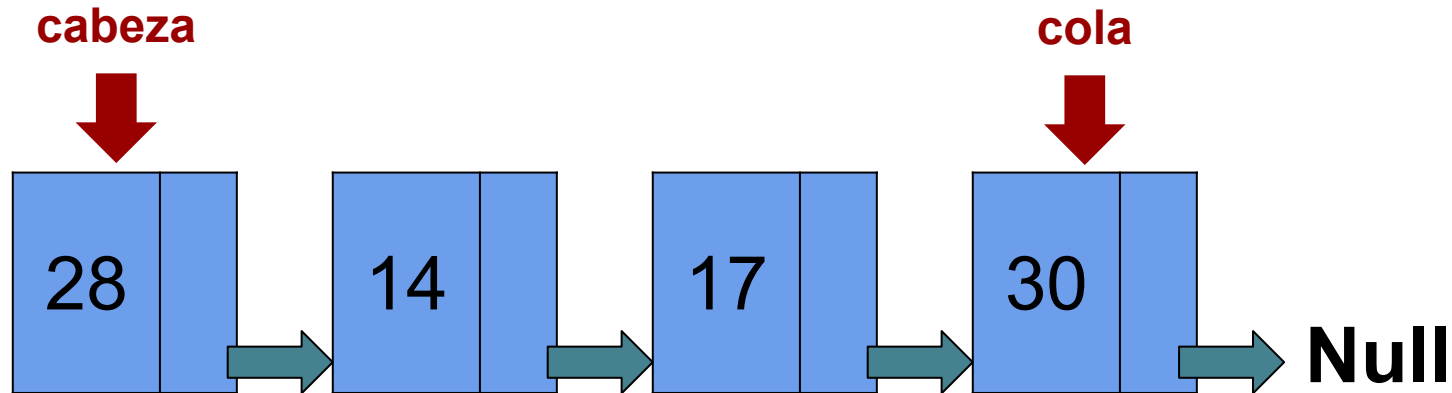


# Operación 2:

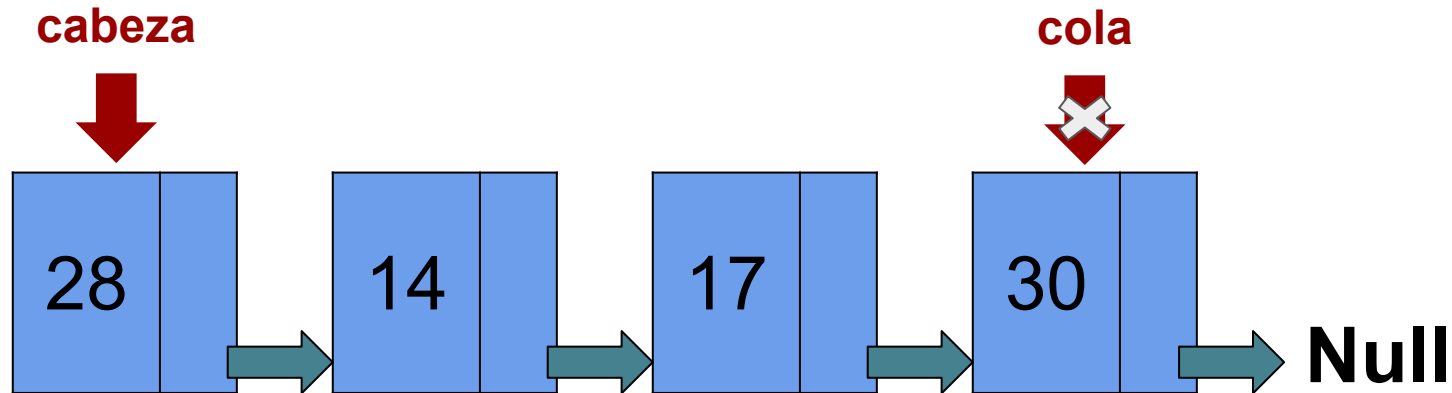
**Remover valores desde la cola de una Lista Enlazada**

# Lista enlazada: remover desde la cola

---

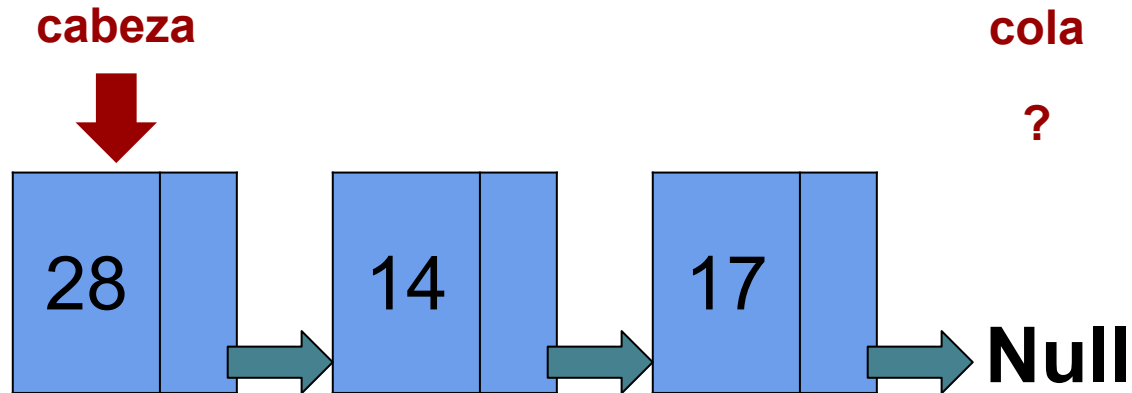


# Lista enlazada: remover desde la cola

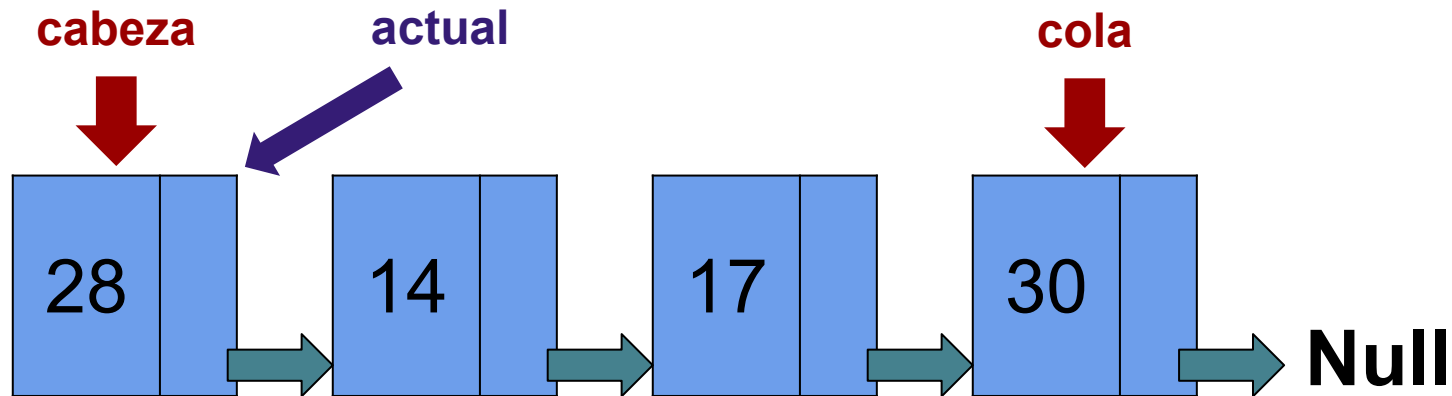


# Lista enlazada: remover desde la cola

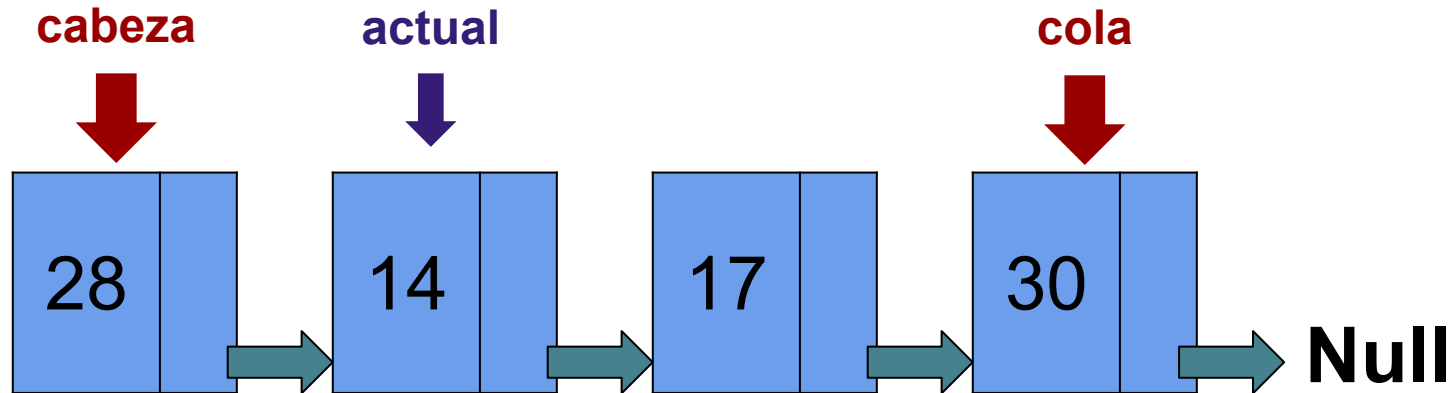
---



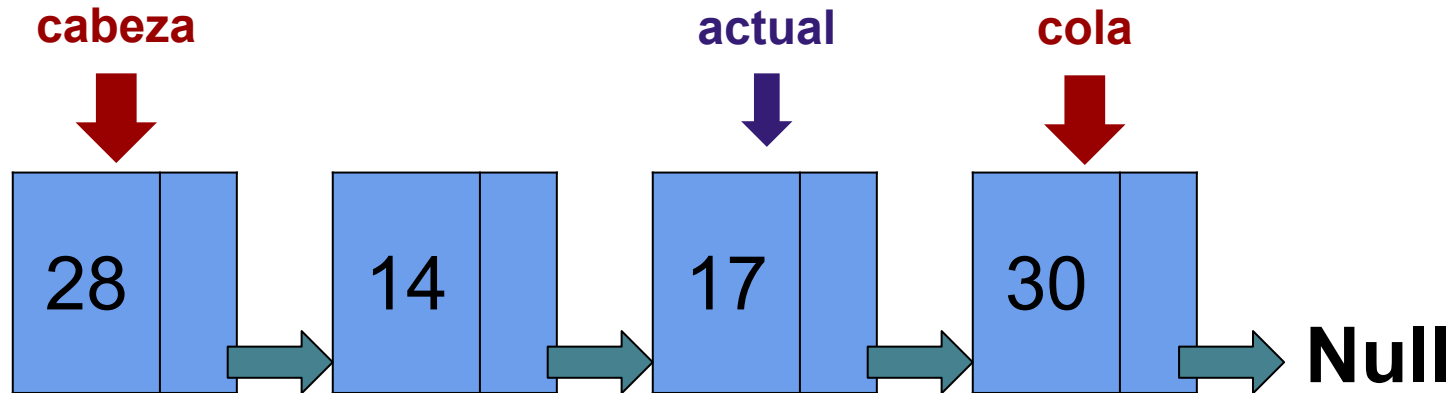
# Lista enlazada: remover desde la cola



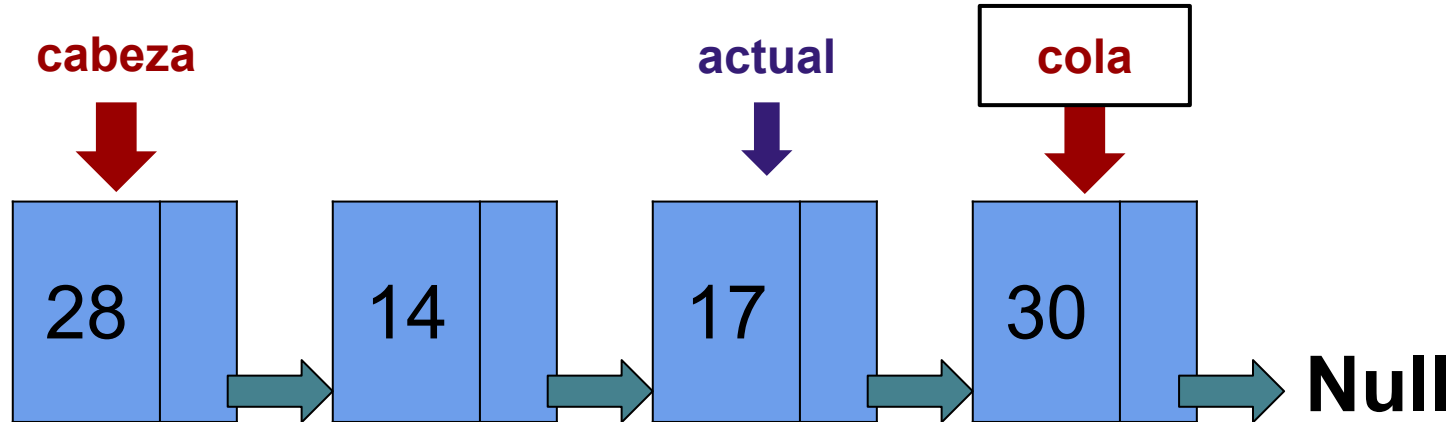
# Lista enlazada: remover desde la cola



# Lista enlazada: remover desde la cola

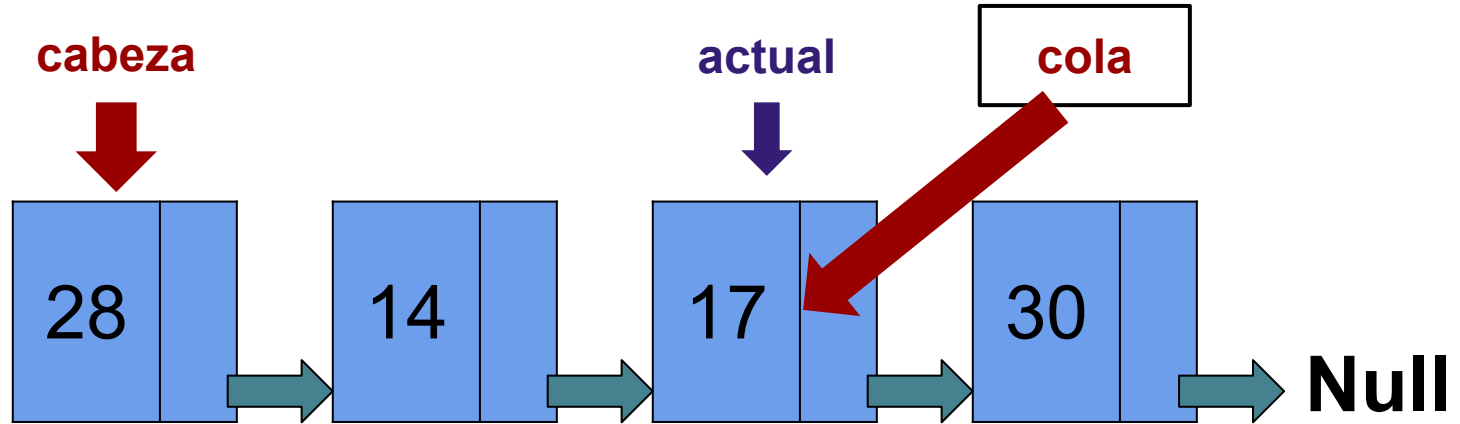


# Lista enlazada: remover desde la cola

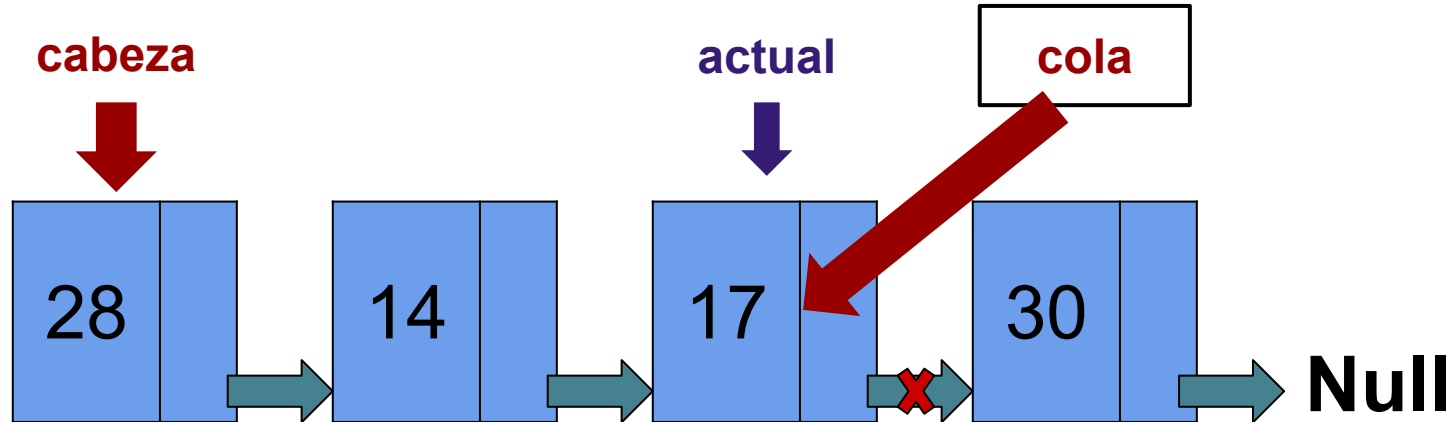




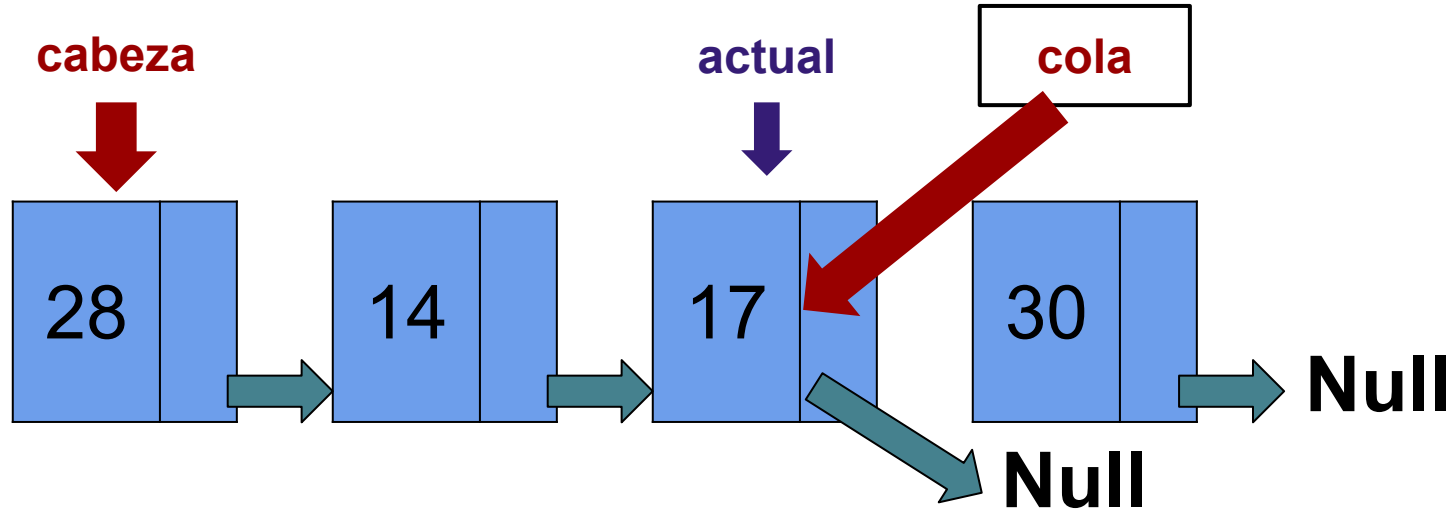
# Lista enlazada: remover desde la cola



# Lista enlazada: remover desde la cola

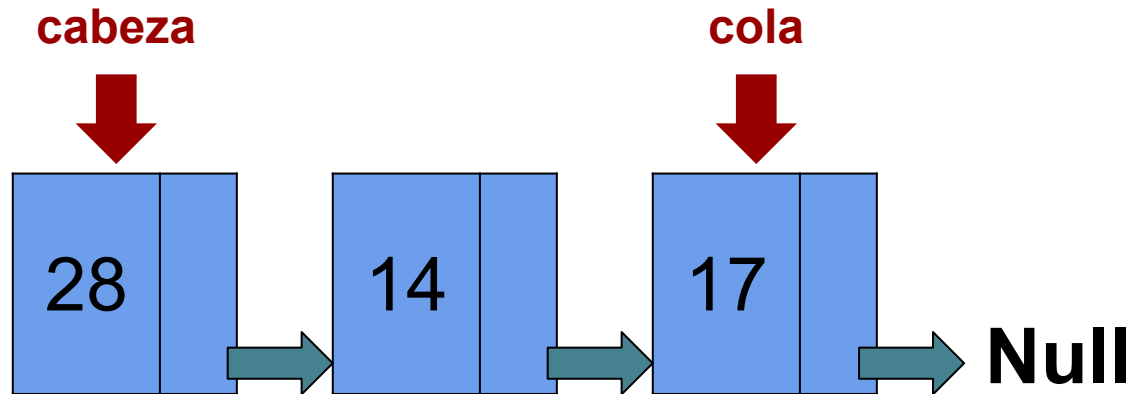


# Lista enlazada: remover desde la cola



# Lista enlazada: remover desde la cola

---



# Lista enlazada: remover desde la cola

---

```
public class ListaEnlazada {  
  
    . . .  
  
    protected Nodo cola = null;  
  
    public void removerDesdeCola () {  
        Nodo actual = cabeza;  
        while ( actual.siguiente != cola ) {  
            actual = actual.siguiente;  
        }  
        cola = actual;  
        actual.siguiente = null;  
    }  
  
}
```

# Lista enlazada: remover desde la cola

---

```
public class ListaEnlazada {  
  
    protected Nodo cola = null;  
  
    public void removerDesdeCola () {  
        if ( cabeza = null ) { // Lista vacía  
            return;  
        } else if ( cabeza.equals(cola) ) { // sólo un elemento en la lista  
            cabeza = null;  
            cola = null;  
        } else {  
            Nodo actual = cabeza;  
            while ( actual.siguiente != cola ) {  
                actual = actual.siguiente;  
            }  
            cola = actual;  
            actual.siguiente = null;  
        }  
    }  
}
```

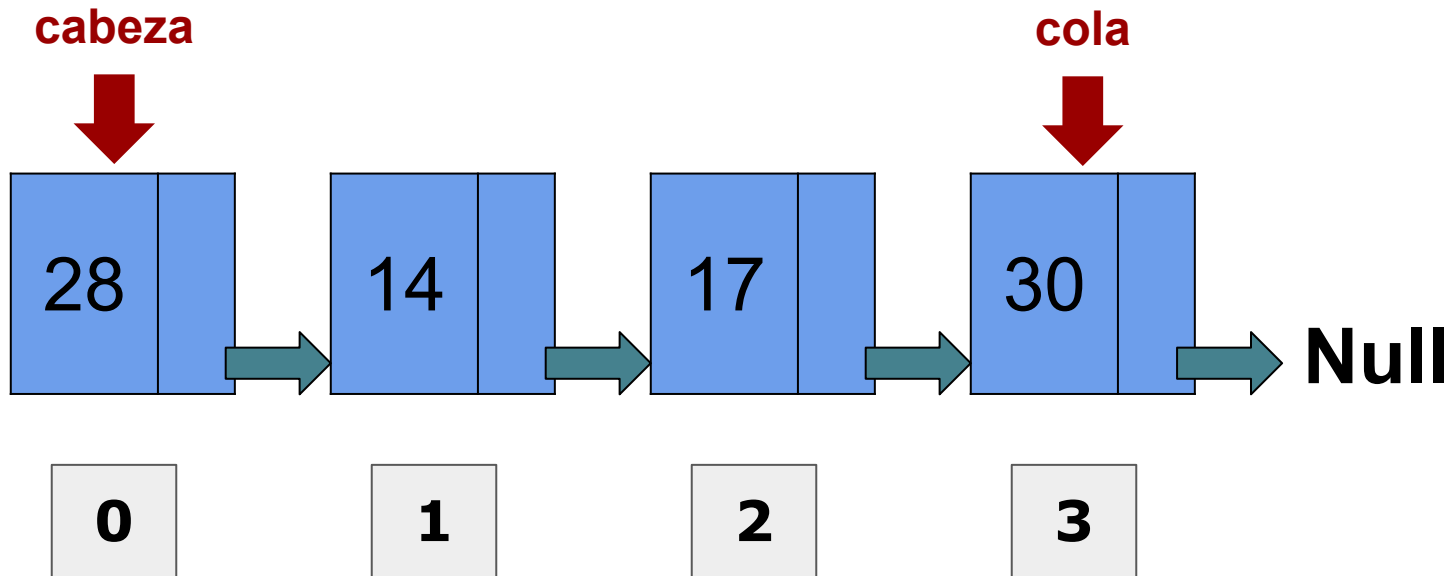


# Operación 3:

**Remover valores desde una posición particular de una Lista Enlazada**

# Lista enlazada: remover valor usando índice

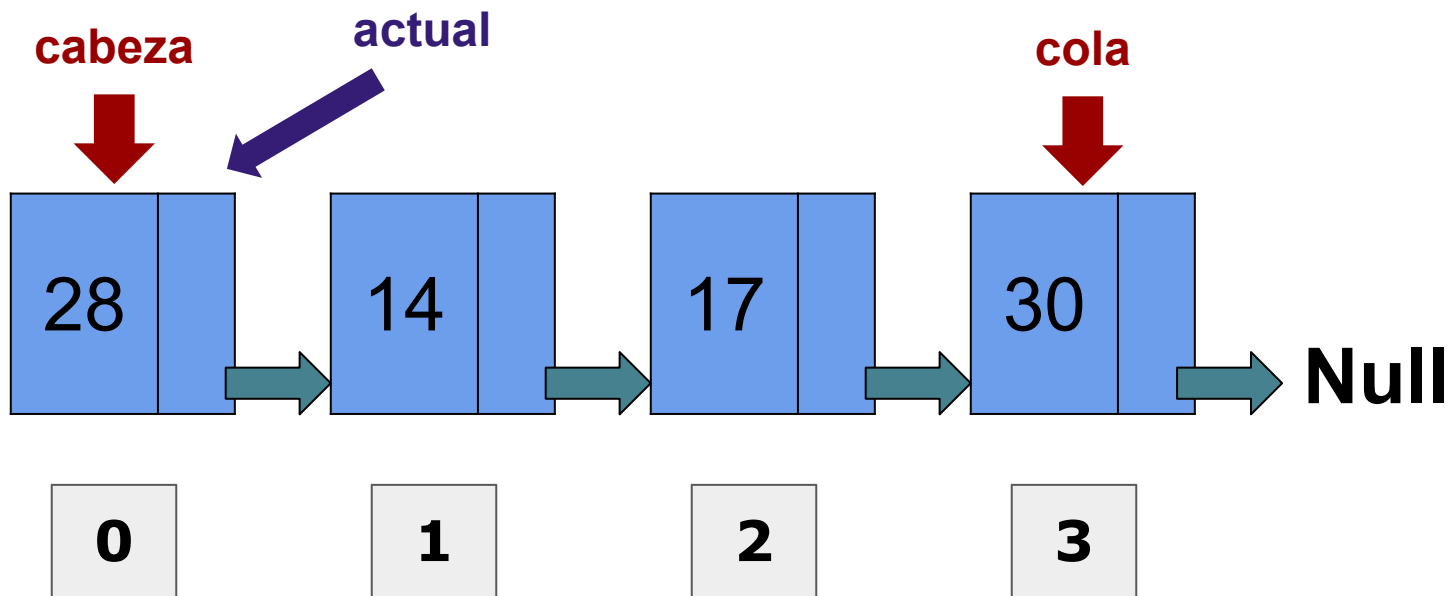
index = 2





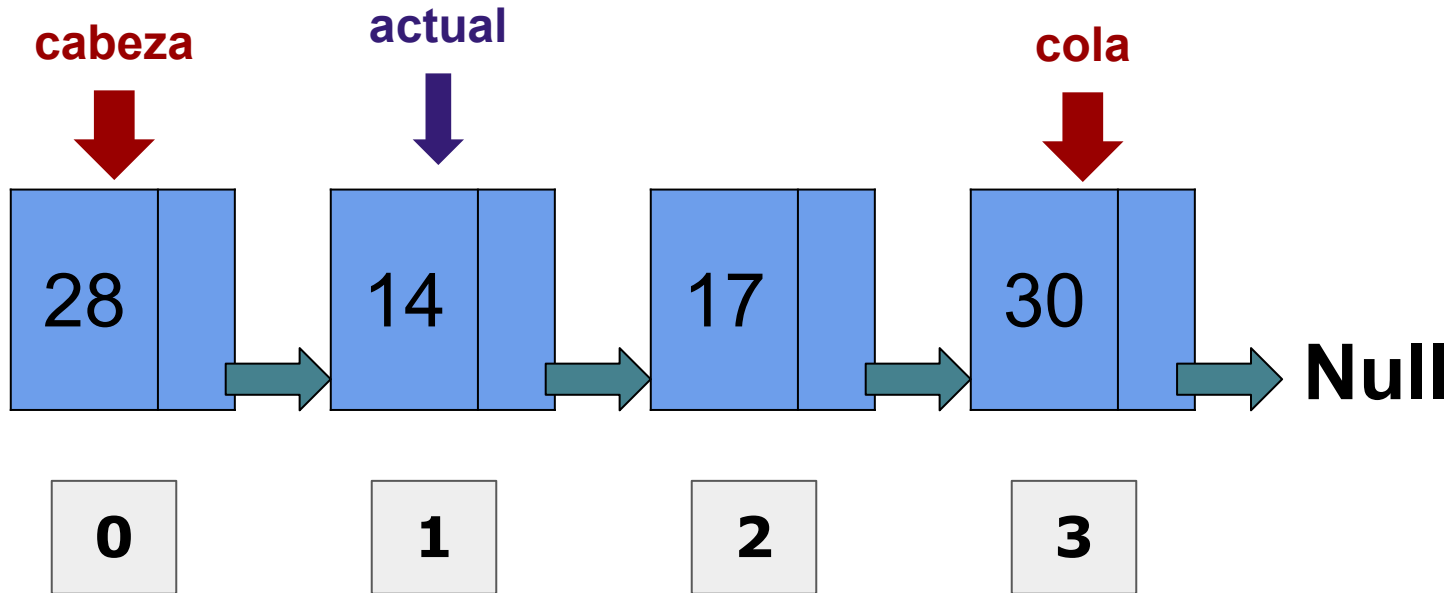
# Lista enlazada: remover valor usando índice

index = 2



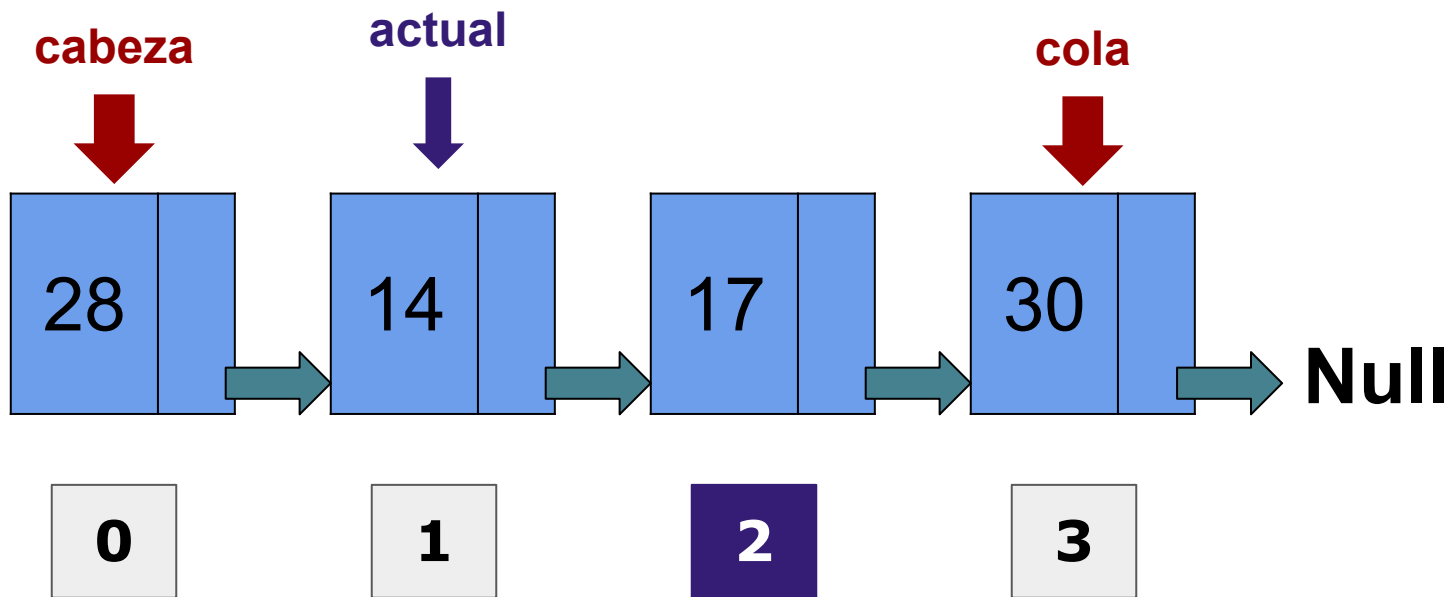
# Lista enlazada: remover valor usando índice

index = 2



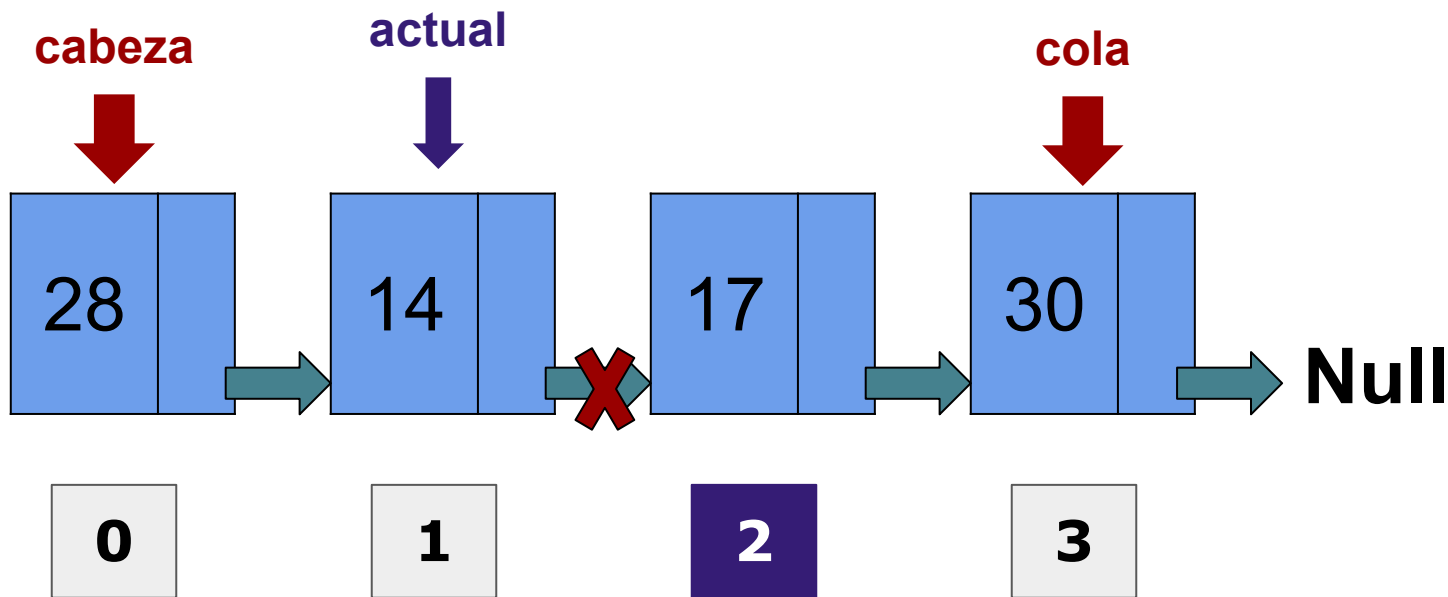
# Lista enlazada: remover valor usando índice

index = 2



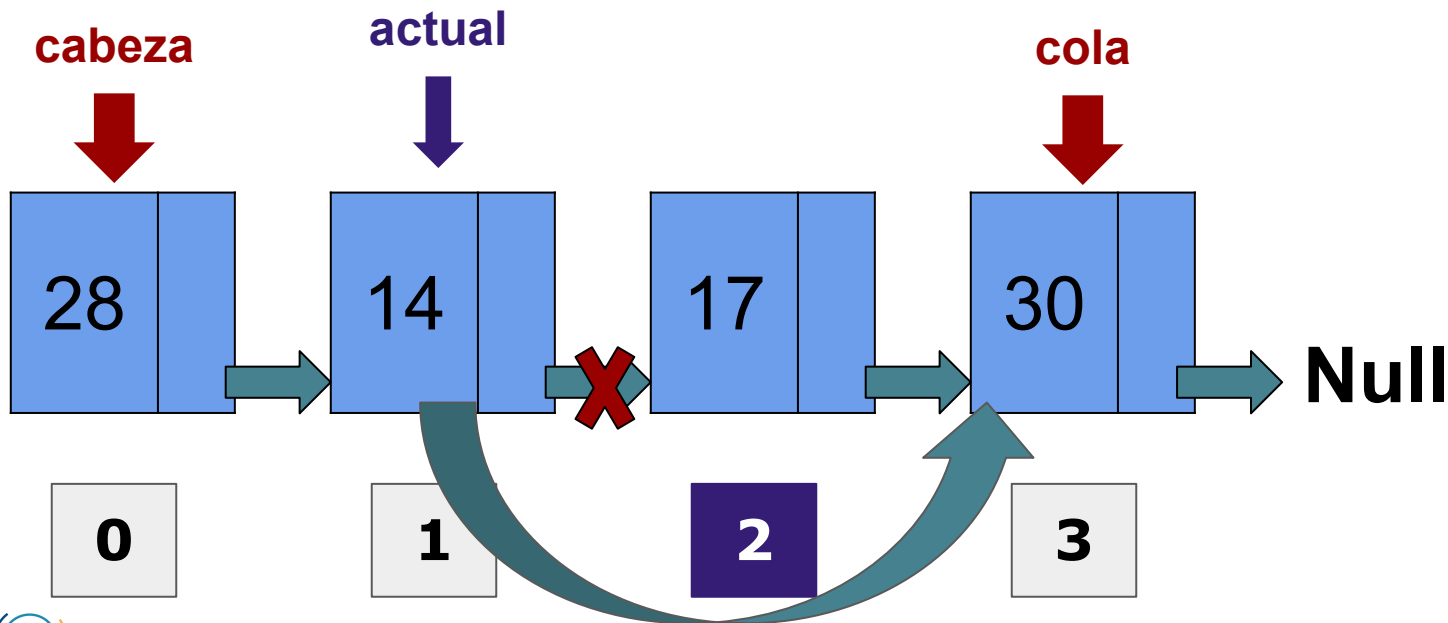
# Lista enlazada: remover valor usando índice

index = 2



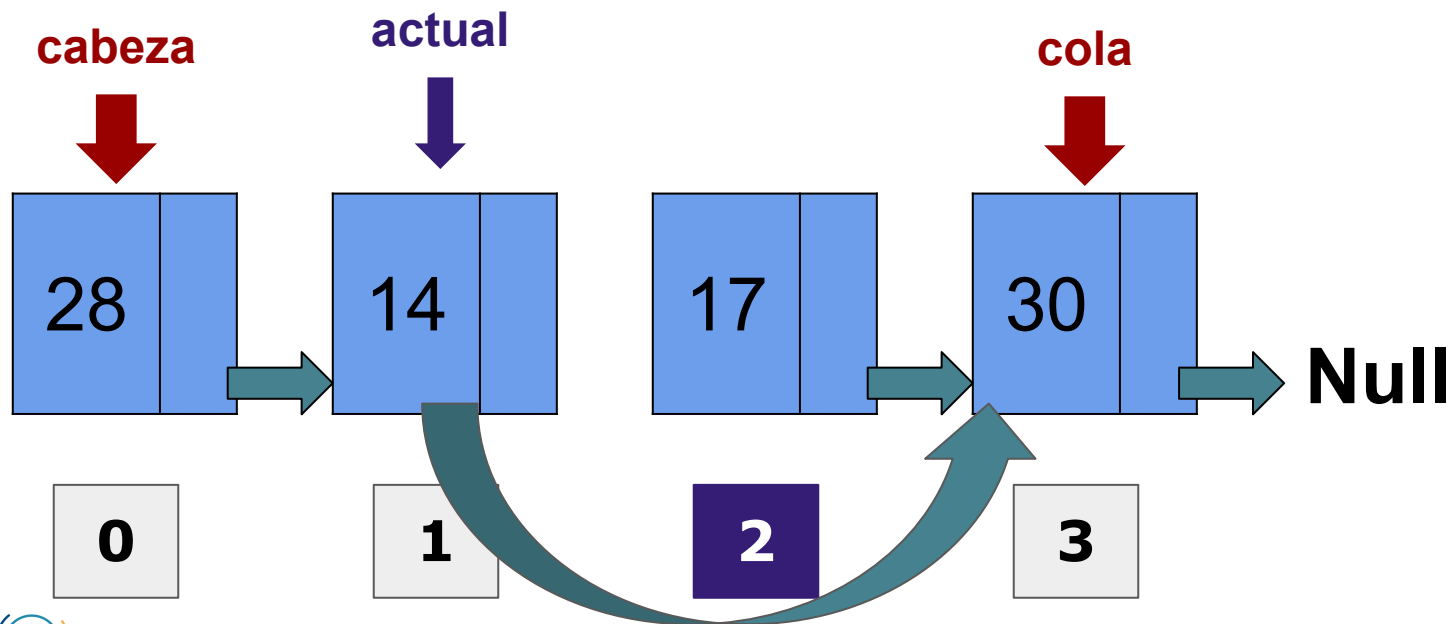
# Lista enlazada: remover valor usando índice

index = 2



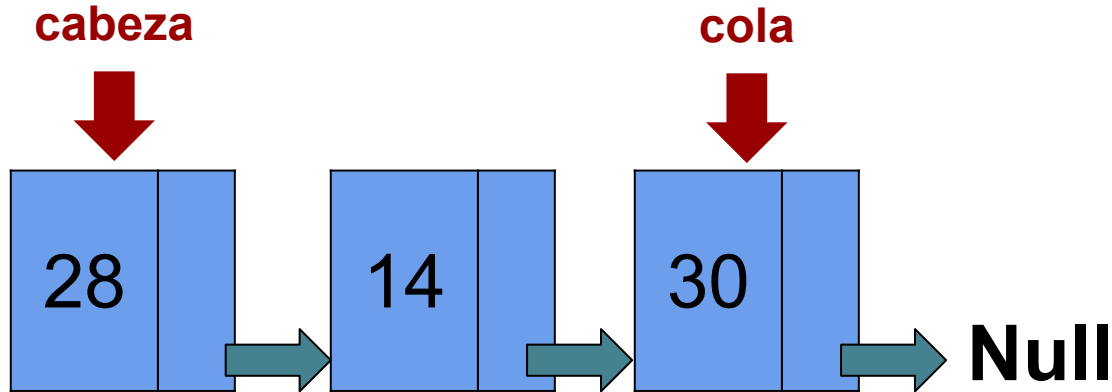
# Lista enlazada: remover valor usando índice

index = 2



# Lista enlazada: remover valor usando índice

---



# Lista enlazada: remover valor usando índice

---

```
public class ListaEnlazada {  
  
    protected Nodo cola = null;  
  
    public void removerEnIndice () {  
        Nodo actual = cabeza;  
        for ( int i = 0; i < index - 1; i++ ) {  
            actual = actual.siguiente;  
        }  
        actual.siguiente = actual.siguiente.siguiente;  
    }  
}
```





# Lista enlazada: remover valor usando índice

---

```
public void removerEnIndice () {  
    if (index < 0) {  
        throw new IndexOutOfBoundsException();  
    } else if ( index == 0 ) {  
        removerDesdeFrente();  
    } else {  
        Nodo actual = cabeza;  
        for ( int i = 0; i < index - 1; i++ ) {  
            if ( actual == null ) {  
                throw new IndexOutOfBoundsException();  
            }  
            actual = actual.siguiente;  
        }  
        actual.siguiente = actual.siguiente.siguiente;  
        if ( actual.siguiente == null ) {  
            cola = actual;  
        }  
    }  
}
```



# **Revisión:**

## **Listas Enlazadas**

# Revisión: Listas Enlazadas

---

1. Serie de Nodos, cada uno enlazado al siguiente
2. Es posible insertar un valor creando un nuevo Nodo y actualizando los enlaces
3. Es posible recuperar/obtener un valor al recorrer la lista
4. Es posible remover un valor desde el frente, según índice y desde la cola



# ICC311

# Estructuras de Datos

Semestre I, 2020

Profesor: Pablo Valenzuela