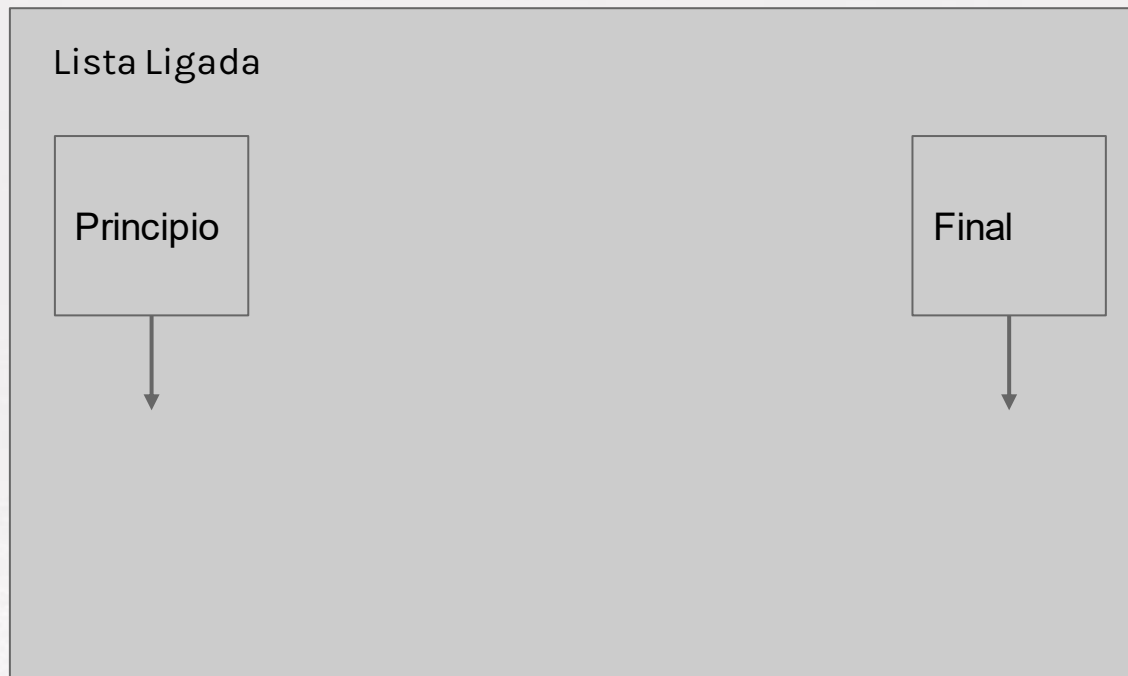


# Cómo funciona una lista ligada

# CREAR

- Inicializa la estructura de datos con los nodos principio y final a null.

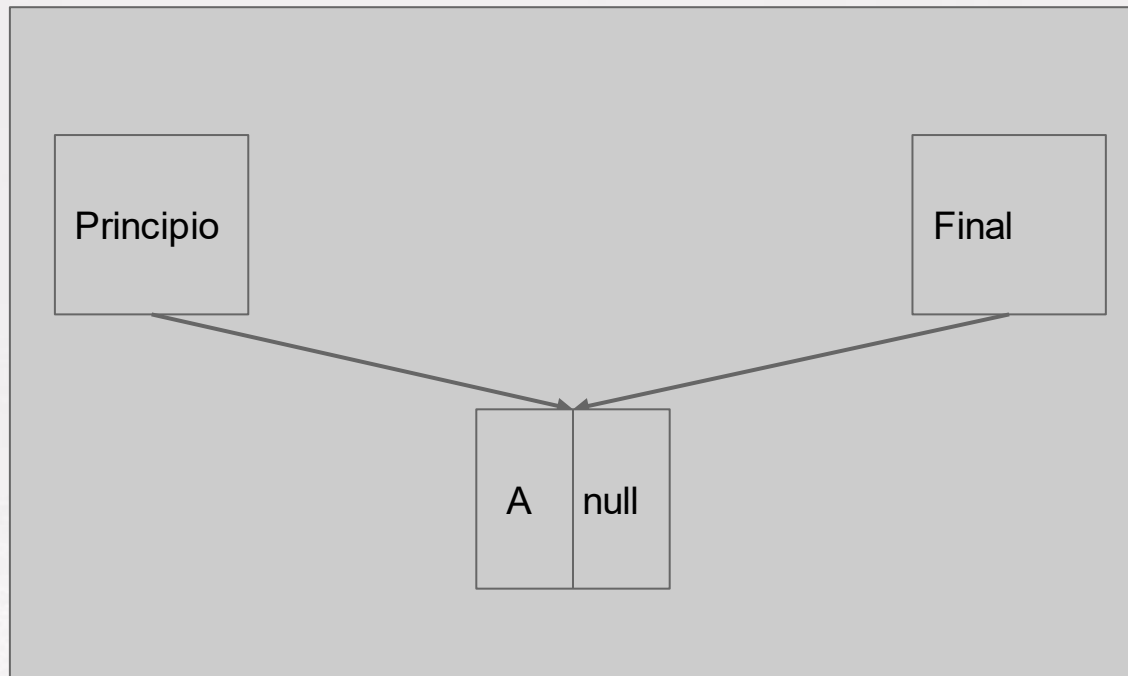
# ESTRUCTURA DE DATOS



# INSERTAR AL PRINCIPIO (LISTA VACÍA)

- Crear un nuevo nodo con el dato - A
- Hacer que primero y último apunten a A

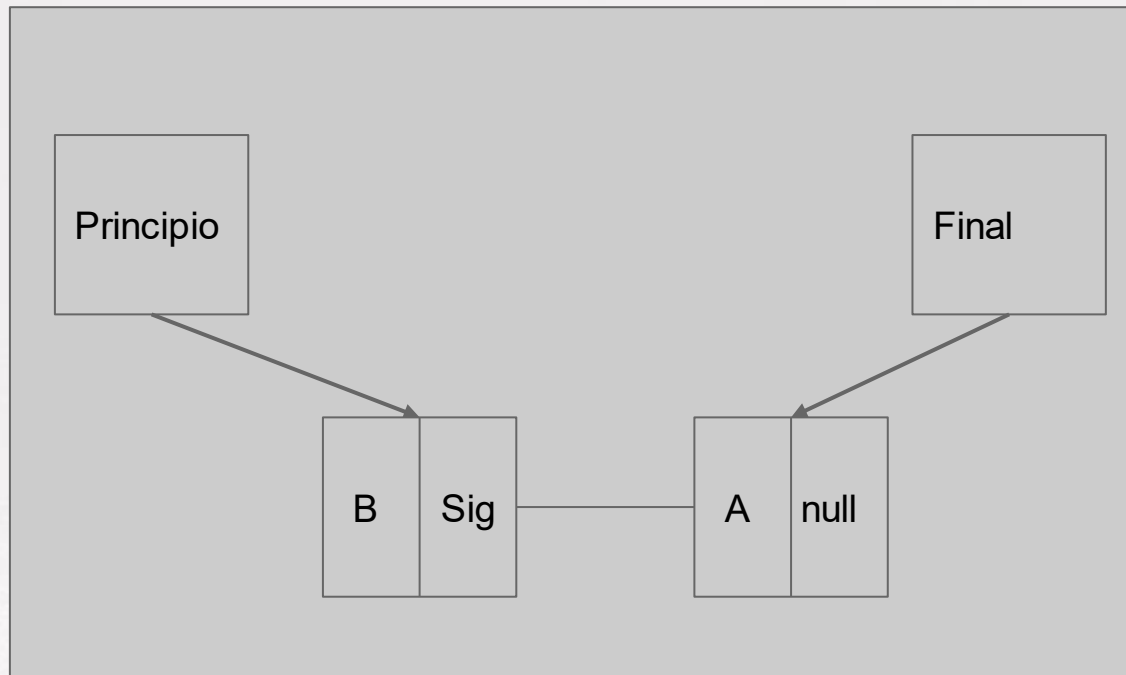
# ESTRUCTURA DE DATOS



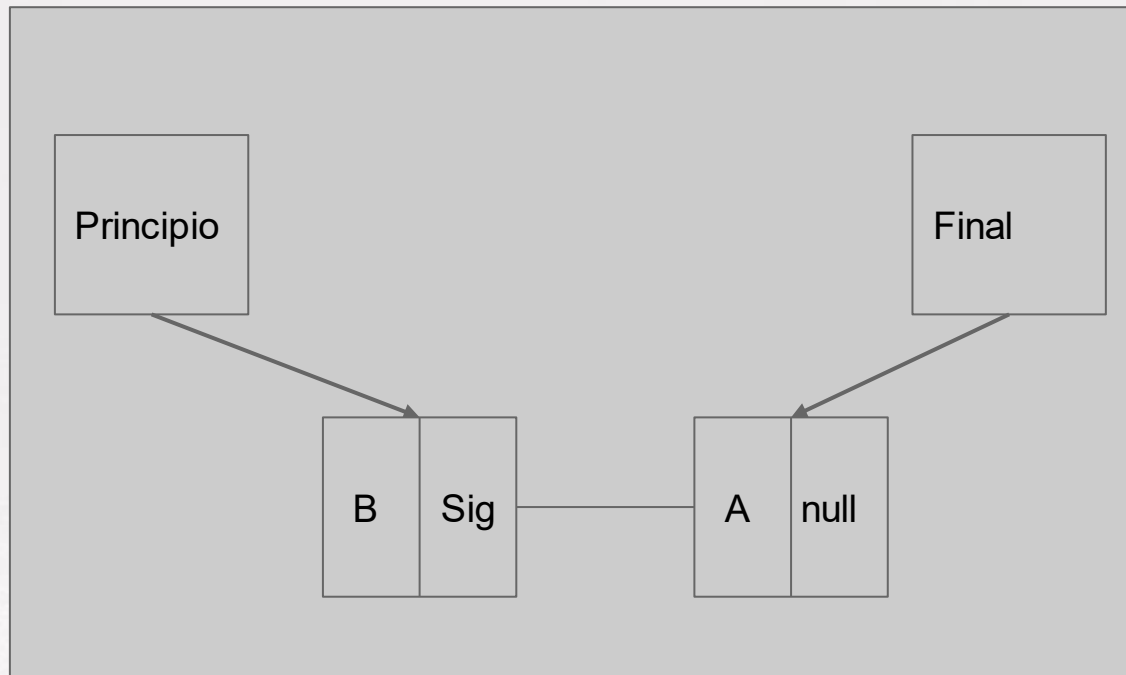
# INSERTAR AL PRINCIPIO (LISTA CON OBJETOS)

- Crear un nuevo nodo con el dato - B
- B.sig = Principio
- Principio = B

# ESTRUCTURA DE DATOS



# ESTRUCTURA DE DATOS





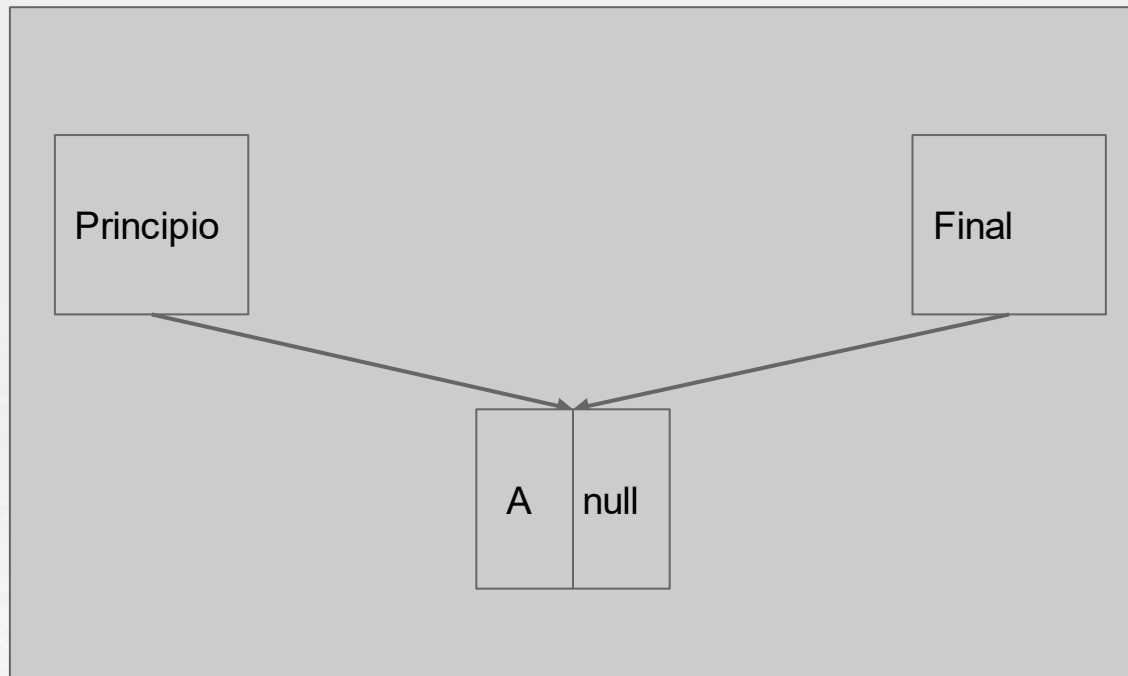
COSTE DE INSERCIÓN AL PRINCIPIO

$O(1)$

# INSERTAR AL FINAL (LISTA VACÍA)

- Crear un nuevo nodo con el dato - A
- Hacer que primero y último apunten a A

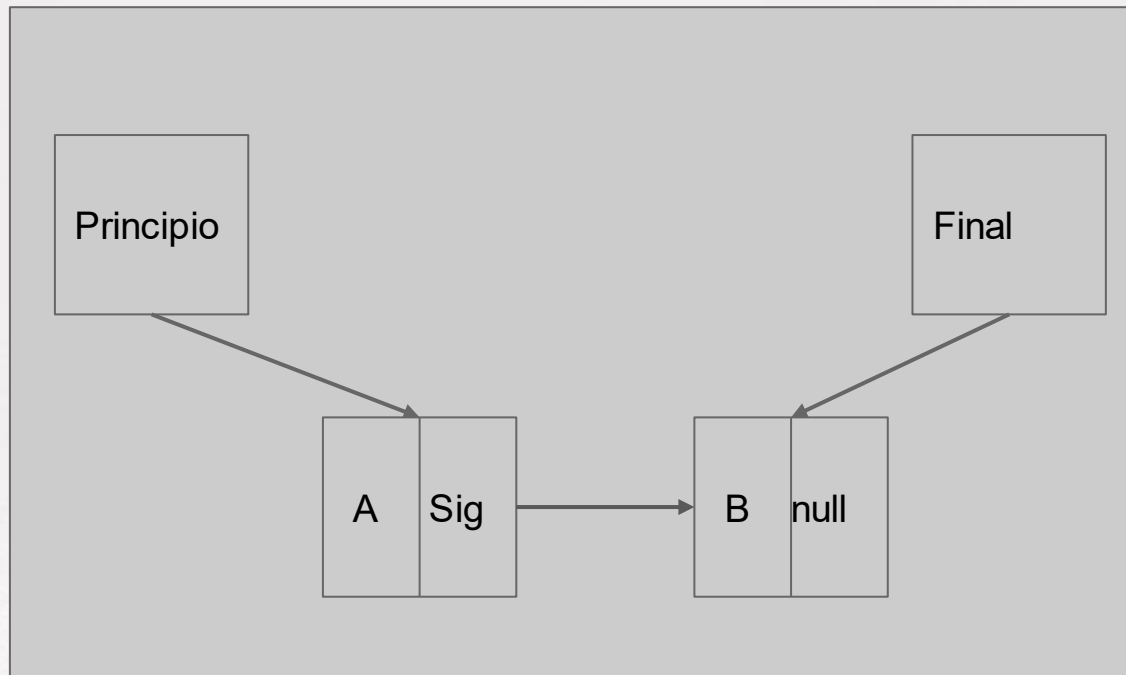
# ESTRUCTURA DE DATOS



# INSERTAR AL FINAL (LISTA CON OBJETOS)

- Crear un nuevo nodo con el dato - B
- `final.sig = B`
- `final = B`

# ESTRUCTURA DE DATOS



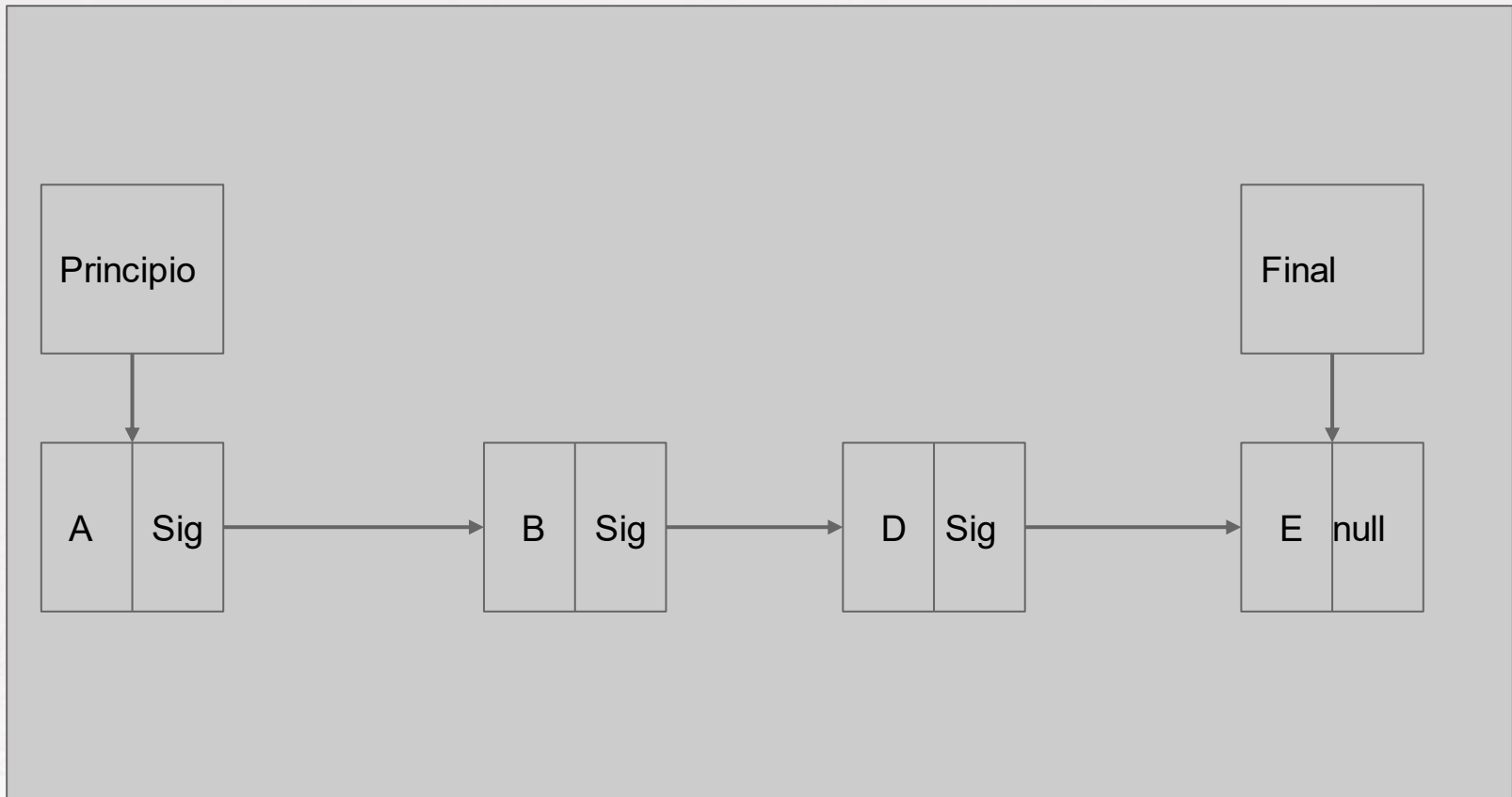
COSTE DE INSERCIÓN AL FINAL

$O(1)$

# INSERTAR EN POSICIÓN

- Crear un nuevo nodo con el dato - A
- Recorrer la lista con dos nodos
  - Un se mantendrá en el nodo anterior
  - El otro en el nodo que se está comprobando si es el que se busca
- Anterior.siguiente = nuevo
- Nuevo.siguiente = actual

# ESTRUCTURA DE DATOS

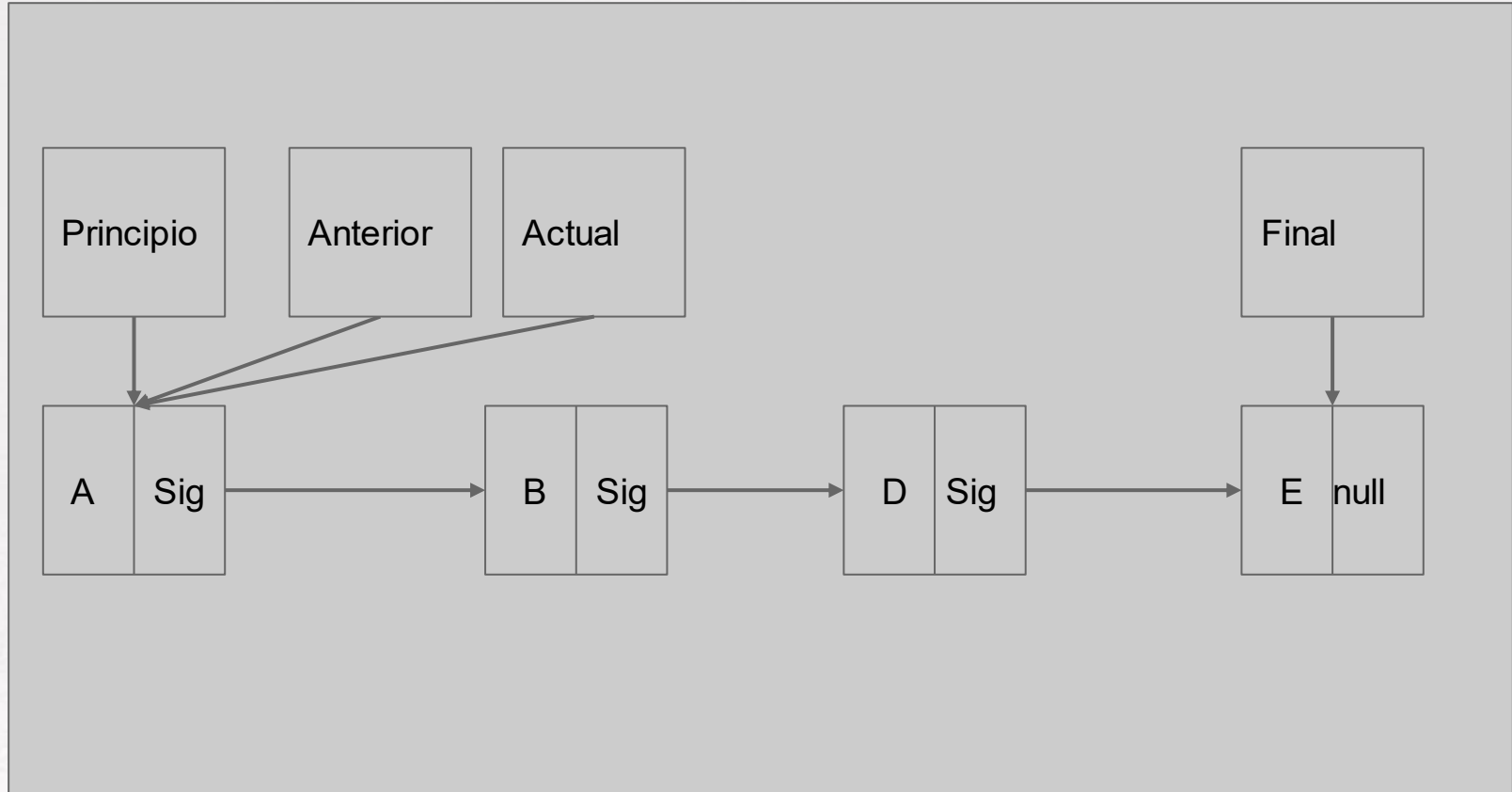




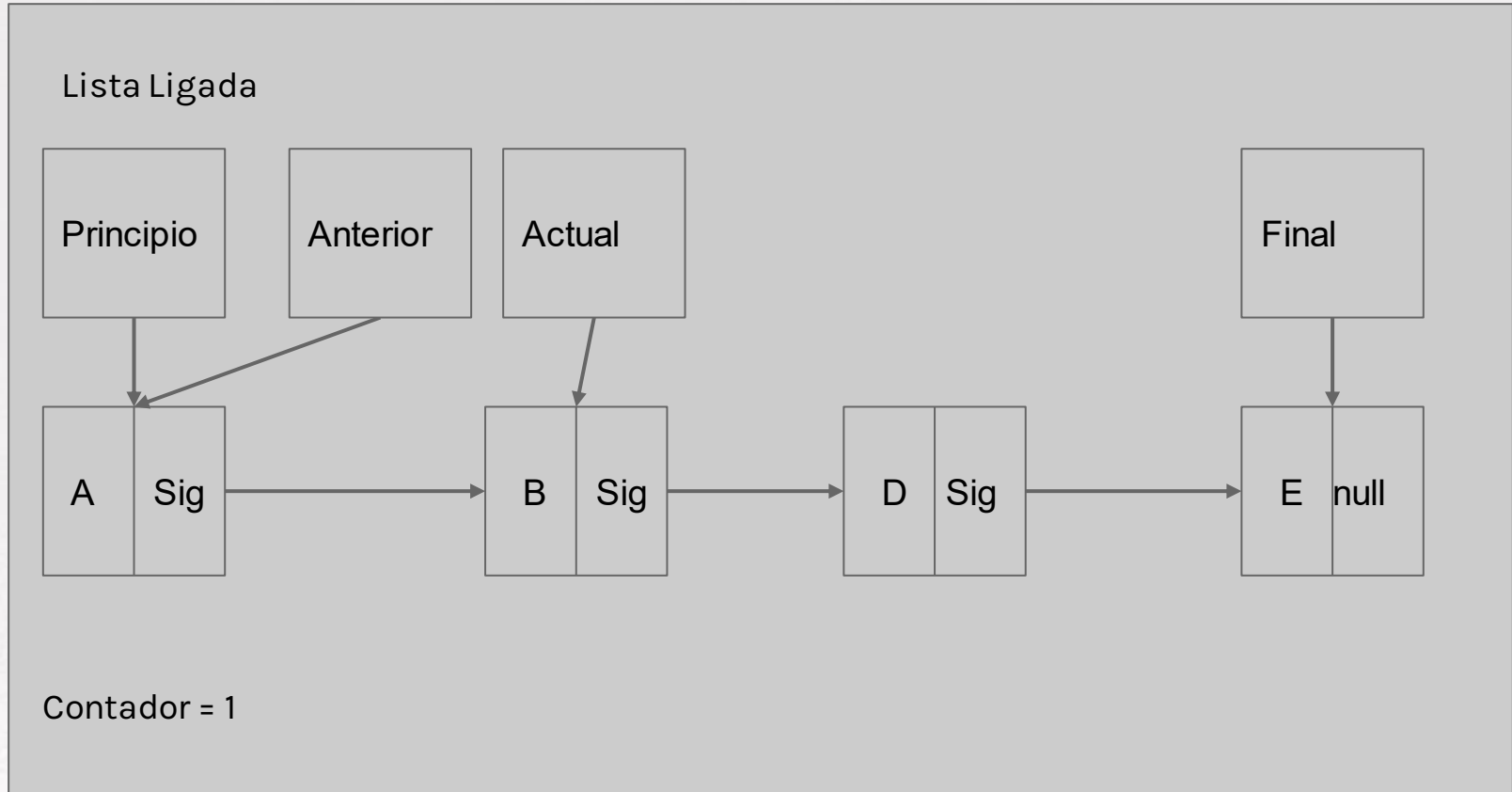
C	null
---	------

Posicion = 2

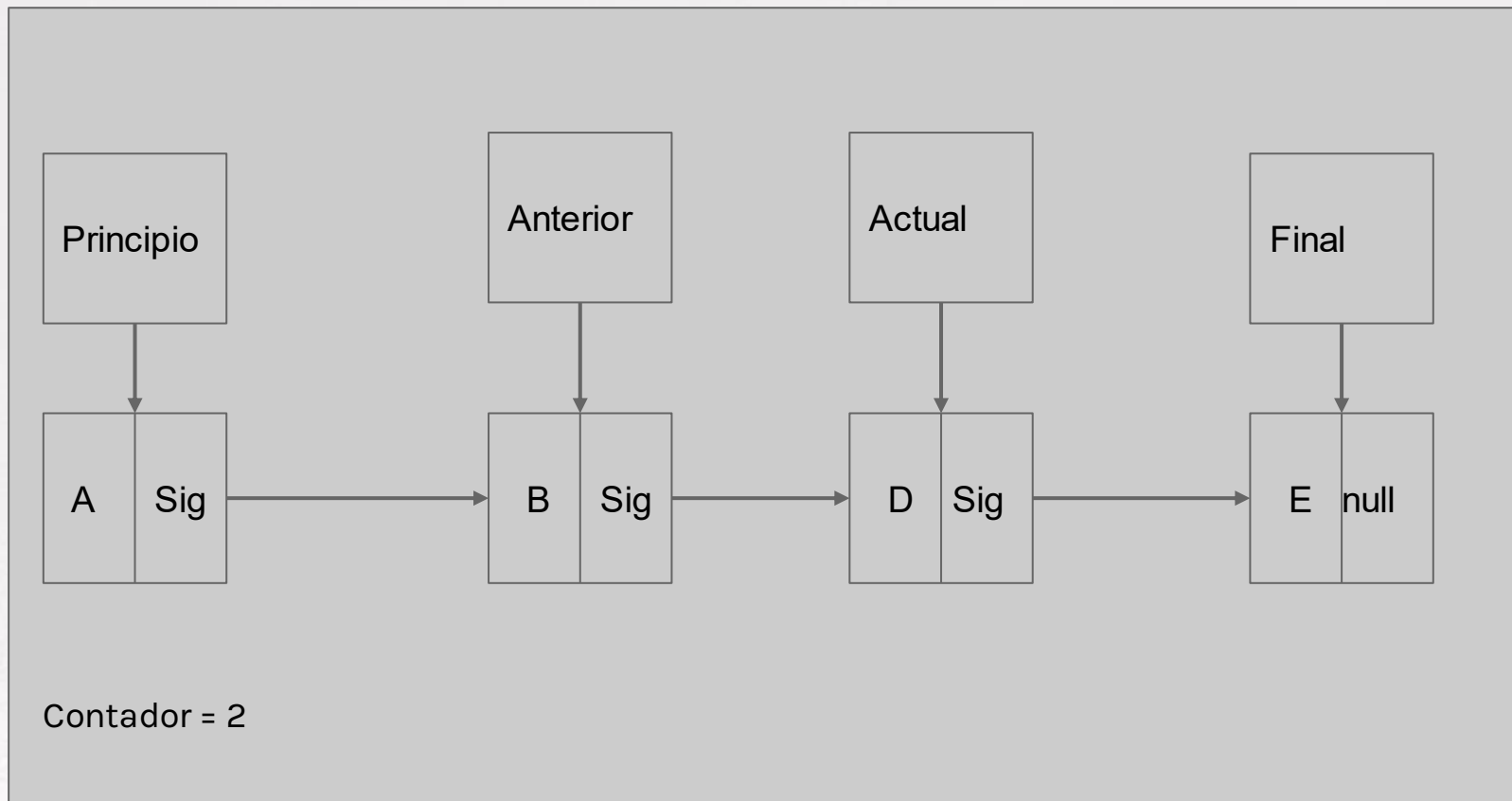
# ESTRUCTURA DE DATOS

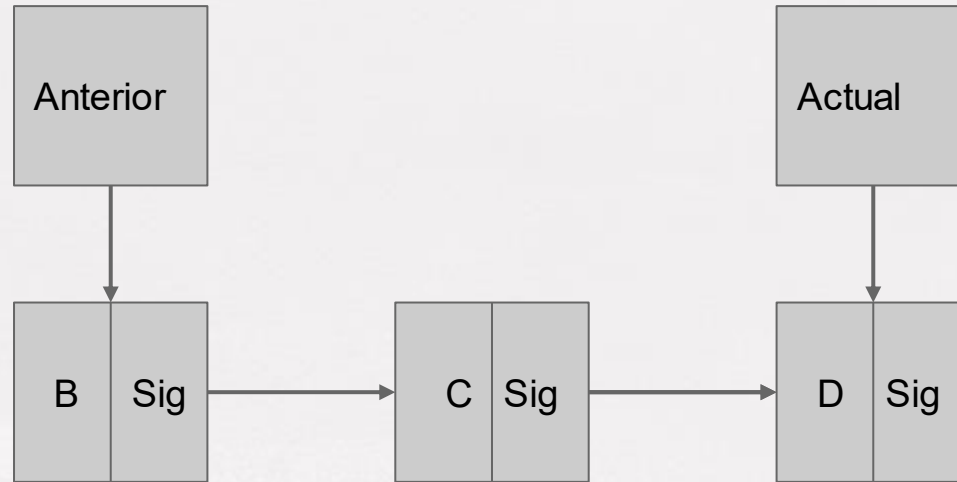


# ESTRUCTURA DE DATOS

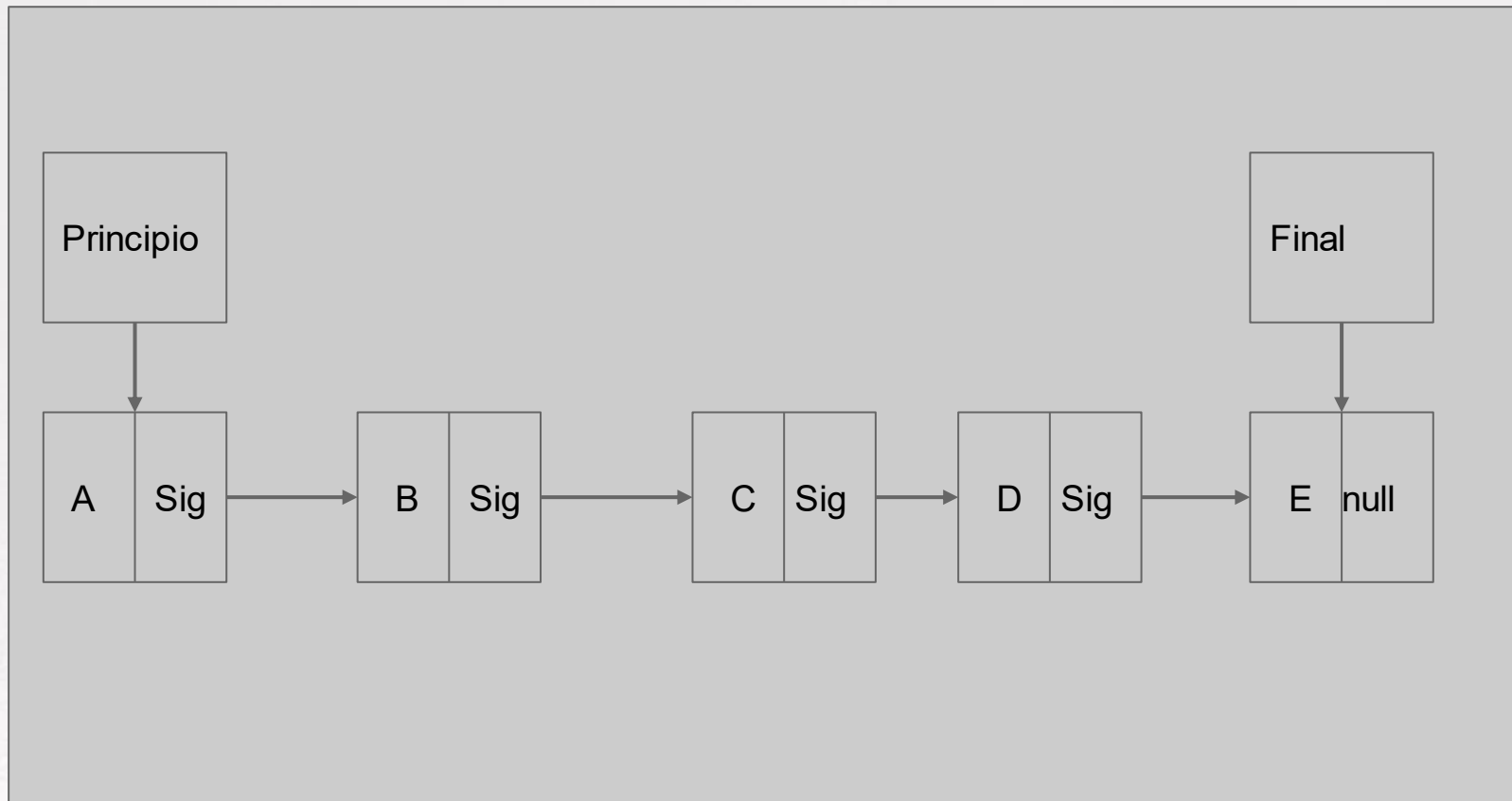


# ESTRUCTURA DE DATOS





# ESTRUCTURA DE DATOS



# COSTE DE INSERCIÓN EN POSICIÓN

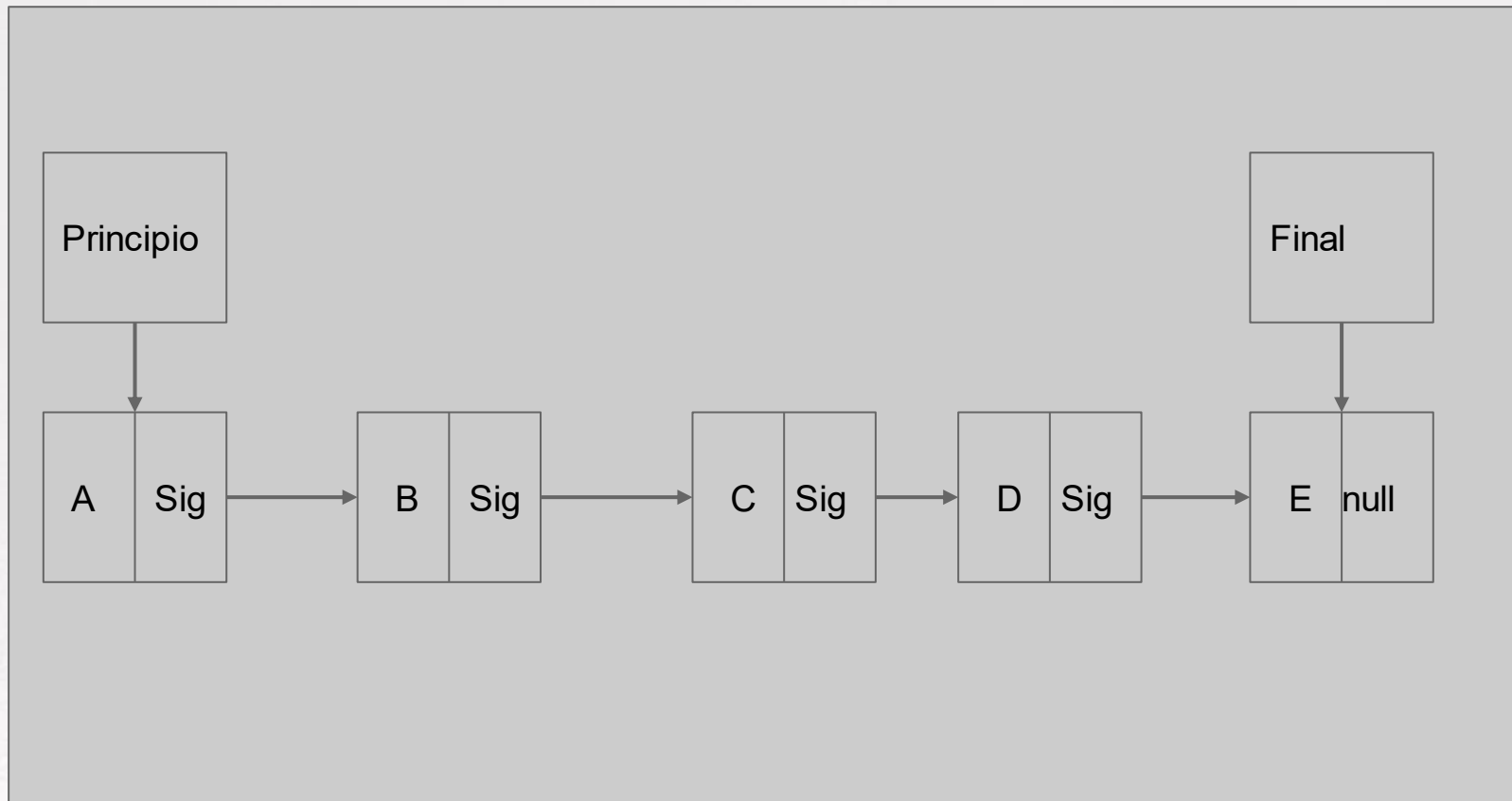
$O(N)$

# SACAR AL PRINCIPIO

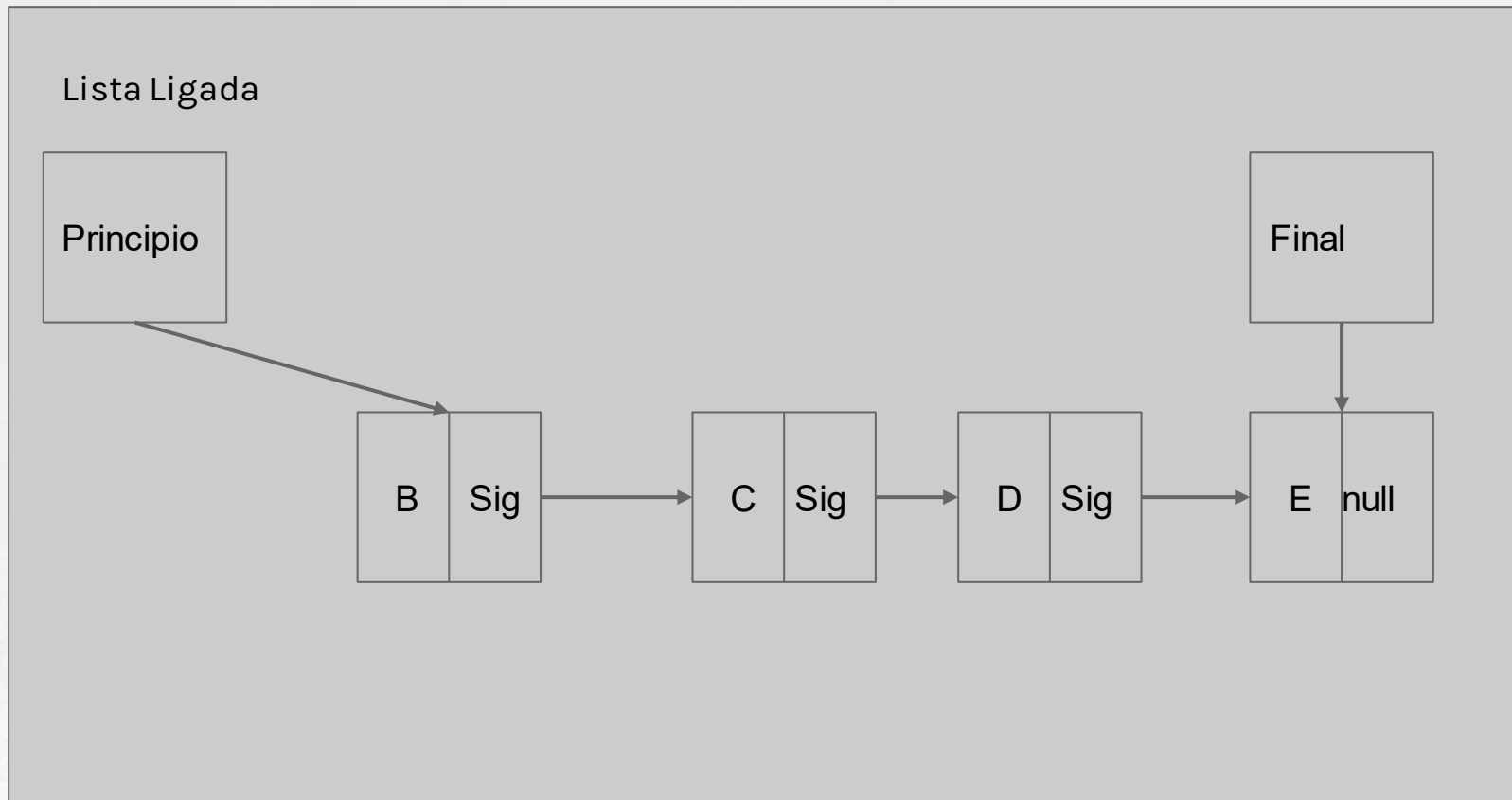
- **Eliminaremos el primer elemento de la lista**
- **Actualizaremos las referencias a primero, y último si fuera necesario**



# ESTRUCTURA DE DATOS



# ESTRUCTURA DE DATOS



COSTE DE SACAR AL PRINCIPIO

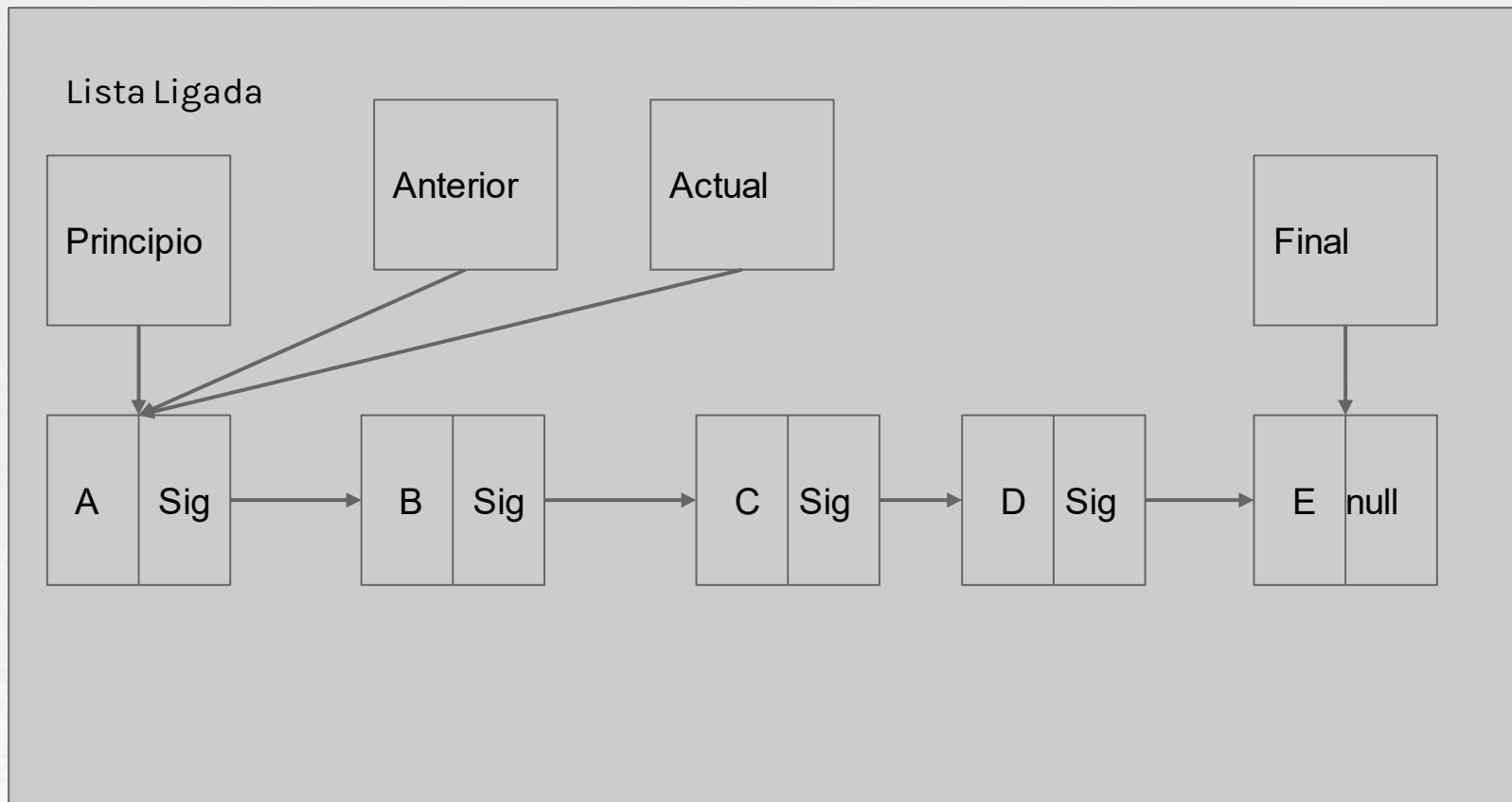
$O(1)$

# SACAR DE UNA POSICIÓN

- Igual que insertar en una posición, nos tendremos que recorrer la lista con dos nodos.

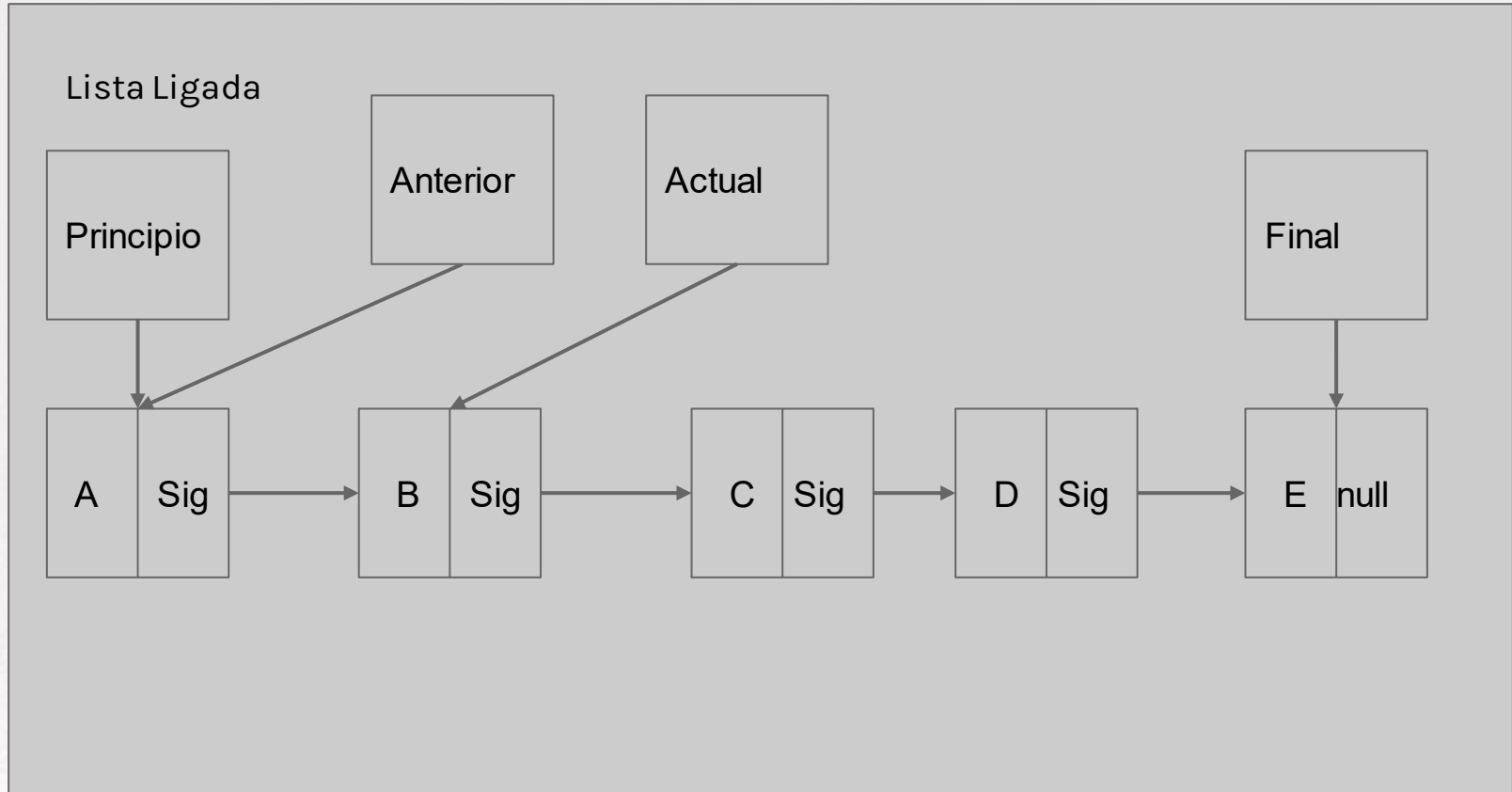
# ESTRUCTURA DE DATOS

Borrar 2



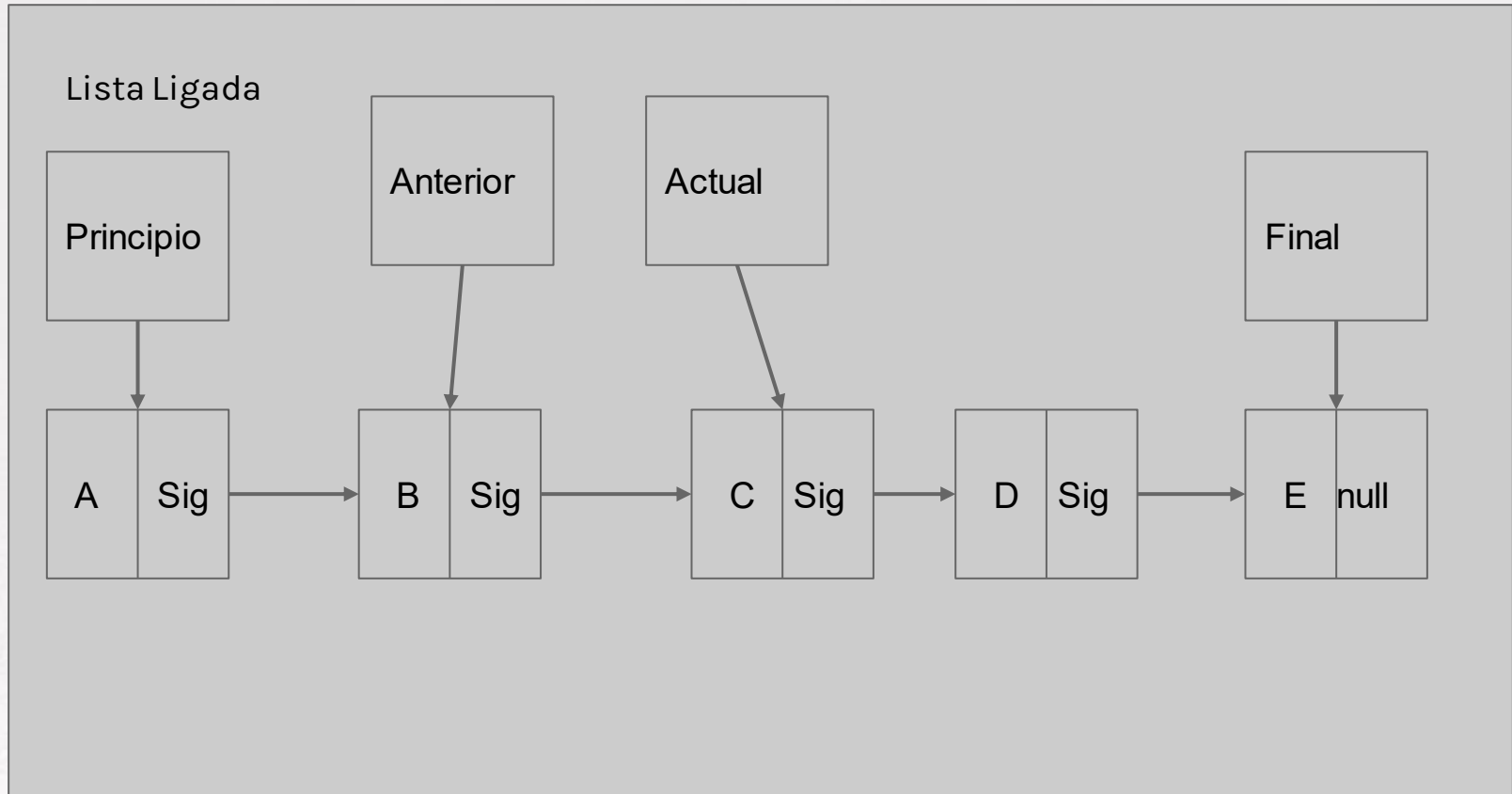
# ESTRUCTURA DE DATOS

Borrar 2

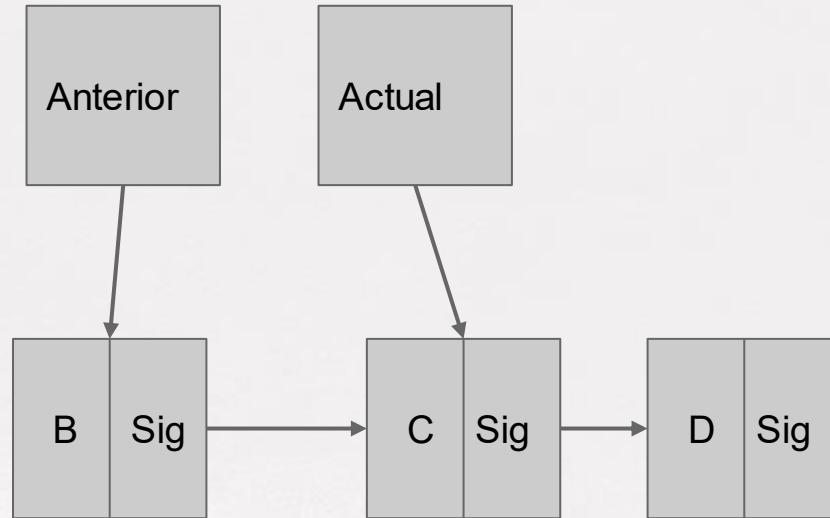


# ESTRUCTURA DE DATOS

Borrar 2

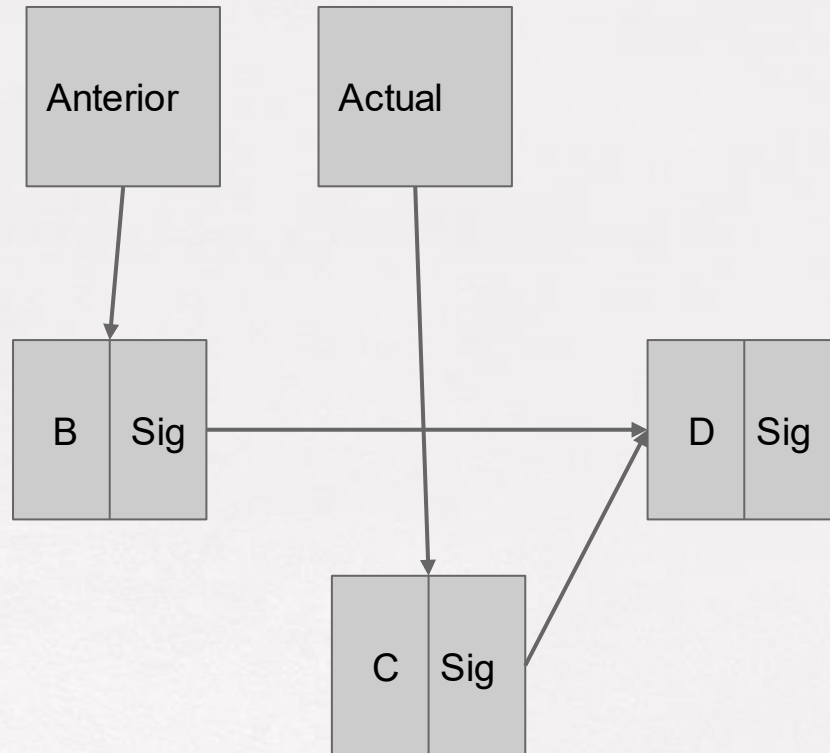


Anterior.sig = actual.siguiente





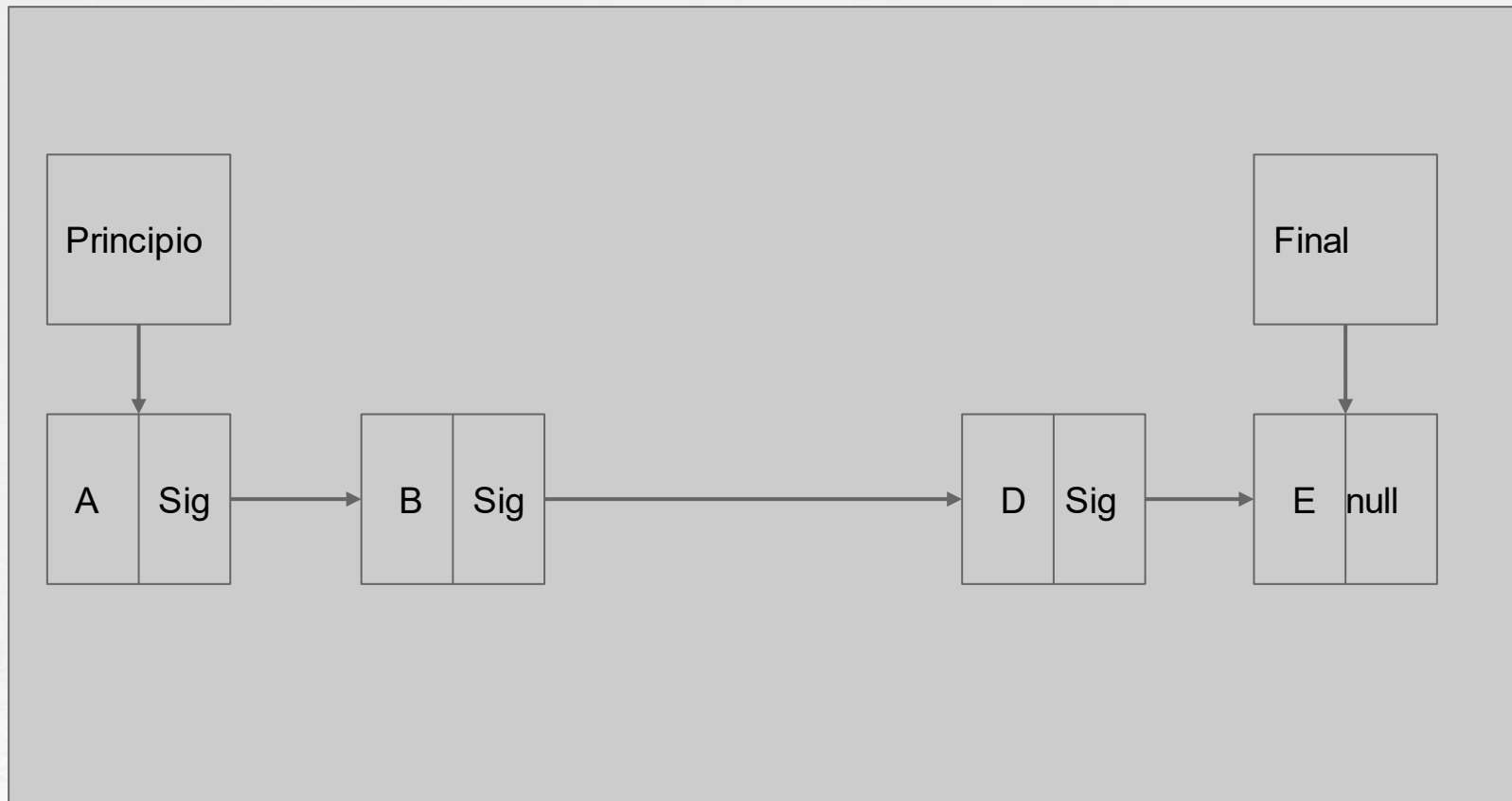
Anterior.sig = actual.siguiente



Anterior.sig = actual.siguiente



# ESTRUCTURA DE DATOS



COSTE DE SACAR UNA POSICIÓN

$O(n)$

## VER EL ELEMENTO DEL PRINCIPIO

Solo necesitaríamos devolver el dato del nodo al que apunta Primero

$$O(1)$$

## VER EL ELEMENTO DEL FINAL

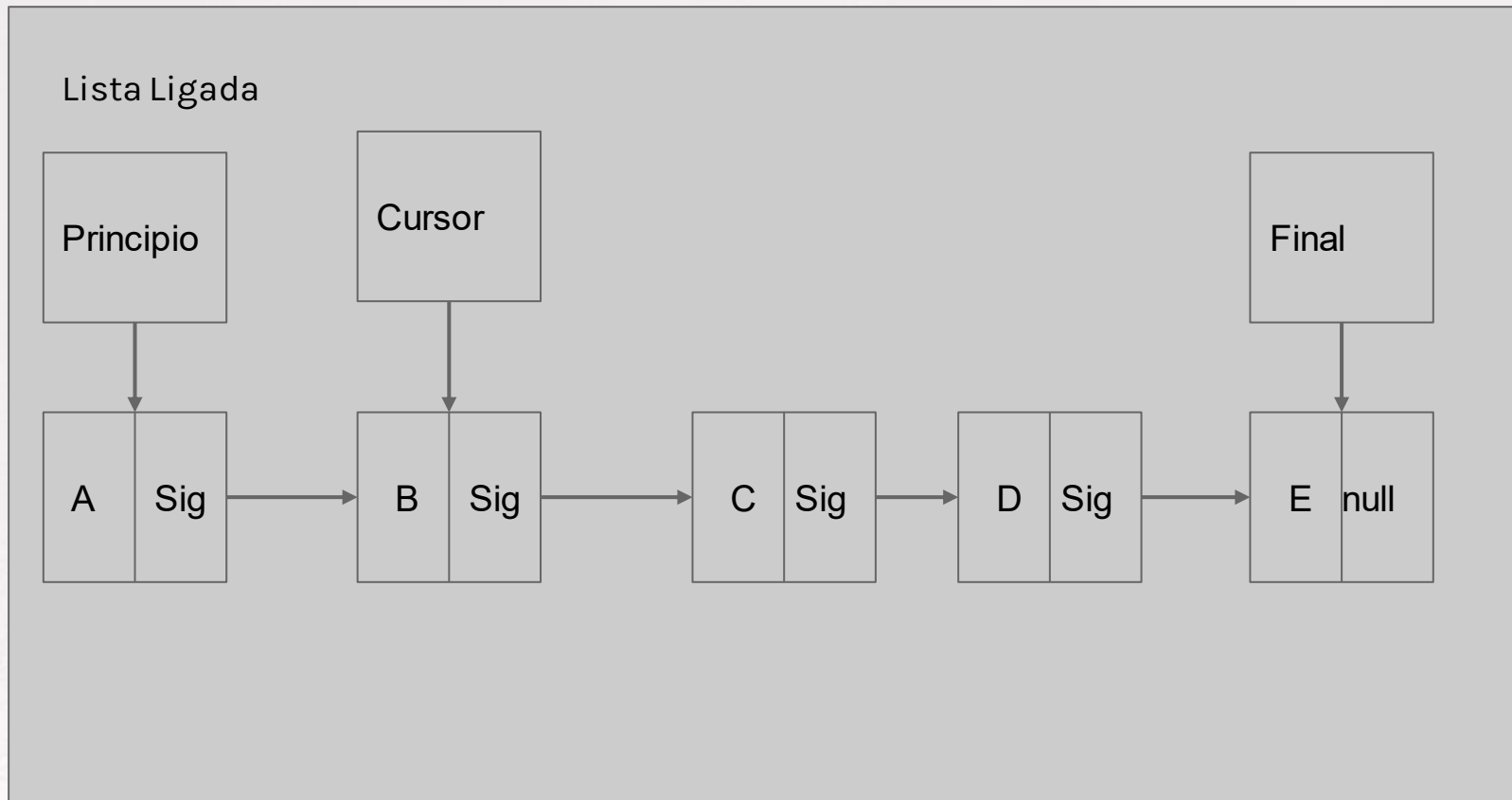
Solo necesitaríamos devolver el dato del nodo al que apunta Final

$$O(1)$$

## VER EL ELEMENTO EN UNA POSICIÓN

**Solo tendríamos que recorreremos la lista con un nodo hasta encontrar el elemento que queremos**

# ESTRUCTURA DE DATOS





COSTE DE SACAR DE UNA POSICIÓN

$O(n)$

Función	Coste
Acceso a posición	$O(n)$
Insertar en posición	$O(n)$
Insertar principio o final	$O(1)$
Eliminar en posición o al final	$O(N)$
Eliminar principio	$O(1)$

**¡GRACIAS!**

**¿ALGUNA PREGUNTA?**