



**UNRN**

Universidad Nacional  
de Río Negro



| unrionegro

# Estructuras de datos

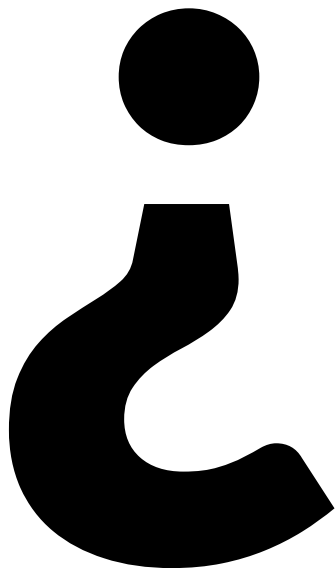
Martín René Vilugron  
martinvilu@unrn.edu.ar

PROGRAMACIÓN 2  
2023

XIV

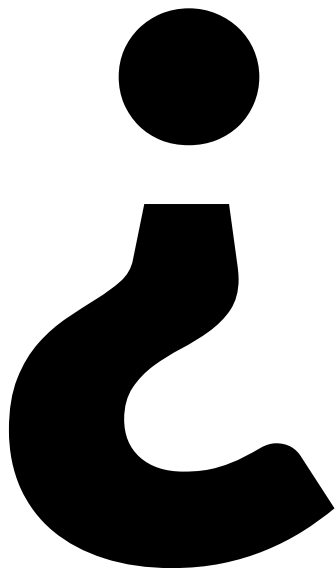


# **El problema del almacenamiento de información**



**Por qué es un  
problema**





**Cuanto 'cuesta'  
cada operación**



# **‘Cuesta’ como complejidad algorítmica $\Theta$**

---

# Secuencias

# Operaciones básicas



# Acceso

# Modificación

# Inserción

# Borrado

# Ordenamiento



**¿Preguntas?**

---

# Array

Conjunto de valores  
consecutivos de tamaño fijo

# Array

**Acceso:  $O(1)$**

**Modificación:  $O(1)$**

**Inserción:  $O(n)$**

**Borrado:  $O(n)$**

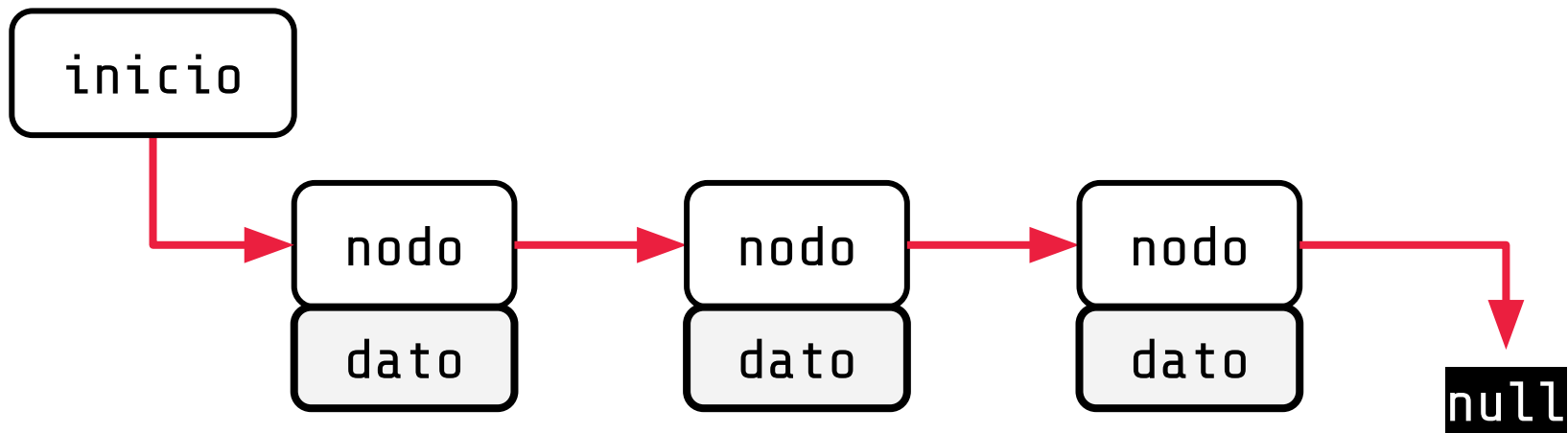
**Ordenamiento:  $O(n^2)$**

**Ampliación:  $O(n)$**



**Hasta acá todo lo de  
siempre**

# Listas Enlazadas



# Listas enlazadas

**Acceso:  $\Omega(1)$  -  $O(n)$**

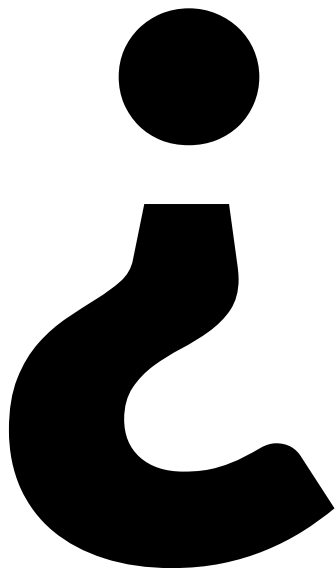
**Modificación:  $\Omega(1)$  a  $O(n)$**

**Inserción:  $\Omega(1)$  a  $O(n)$**

**Borrado:  $\Omega(1)$  a  $O(n)$**

**Ordenamiento:  $O(n^2)$**

**Ampliación:  $\Omega(1)$  a  $O(n)$**



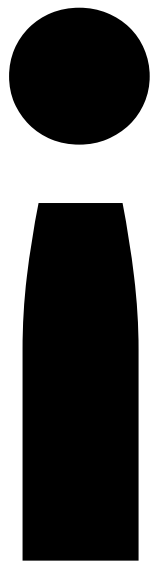
**Por qué**



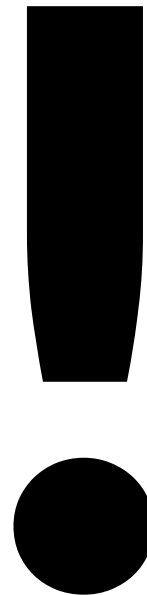
---

# Usos

1. No conocemos el tamaño de la información a procesar
2. Queremos implementar un Arreglo que pueda crecer
3. Gestionar memoria
4. Implementar otras estructuras de datos (más a continuación)

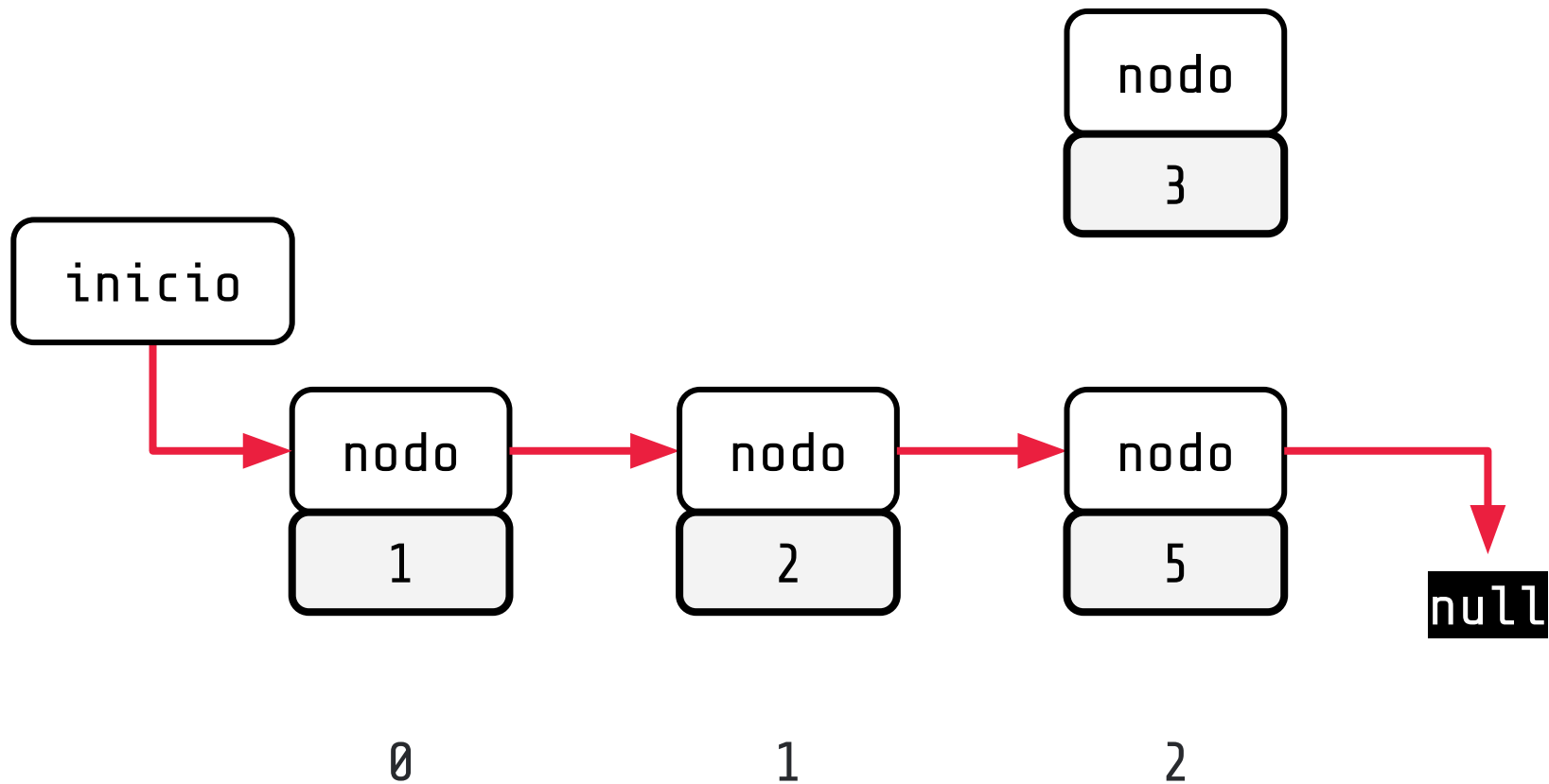


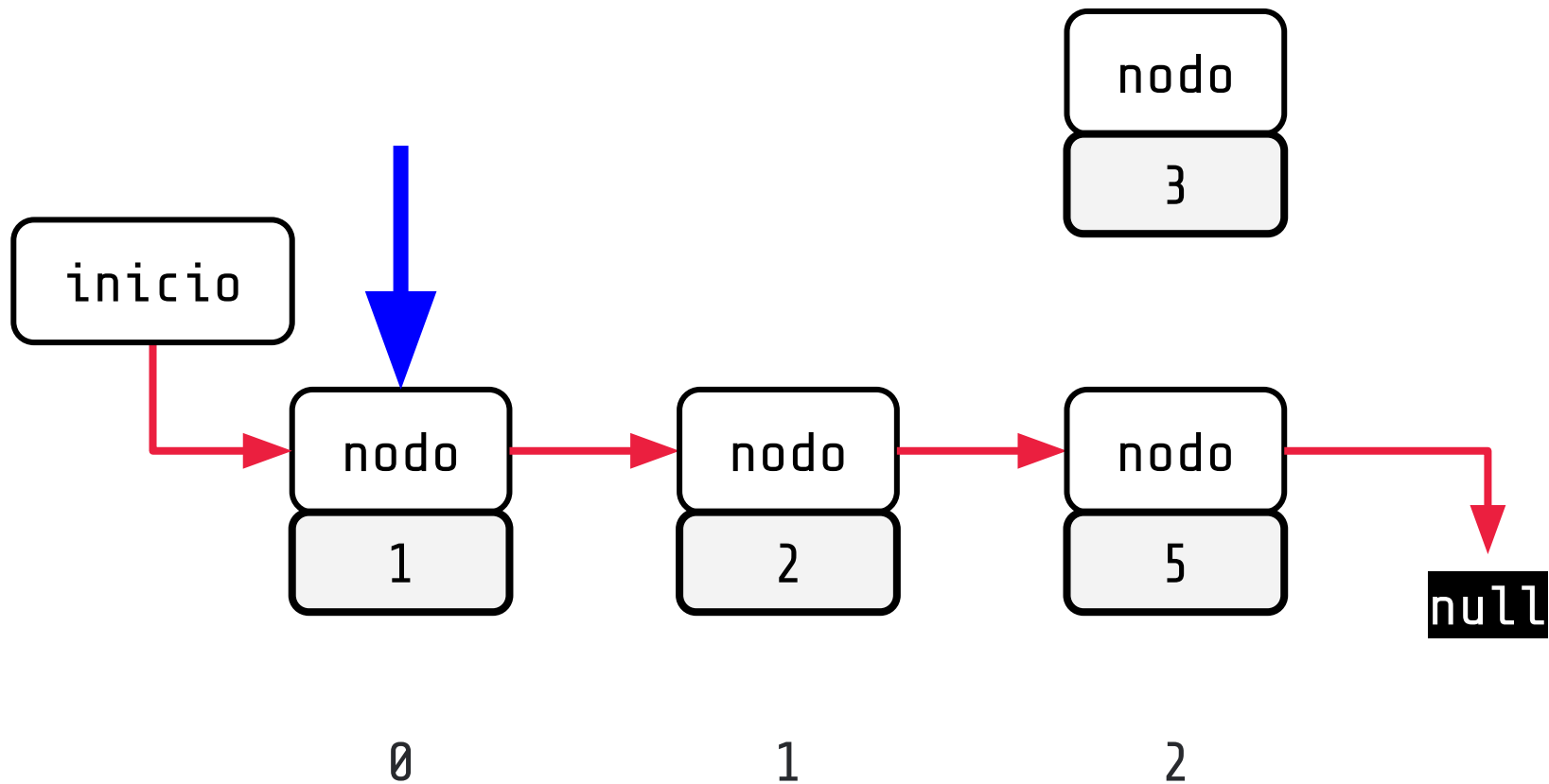
**Solo podemos ir  
hacia adelante**

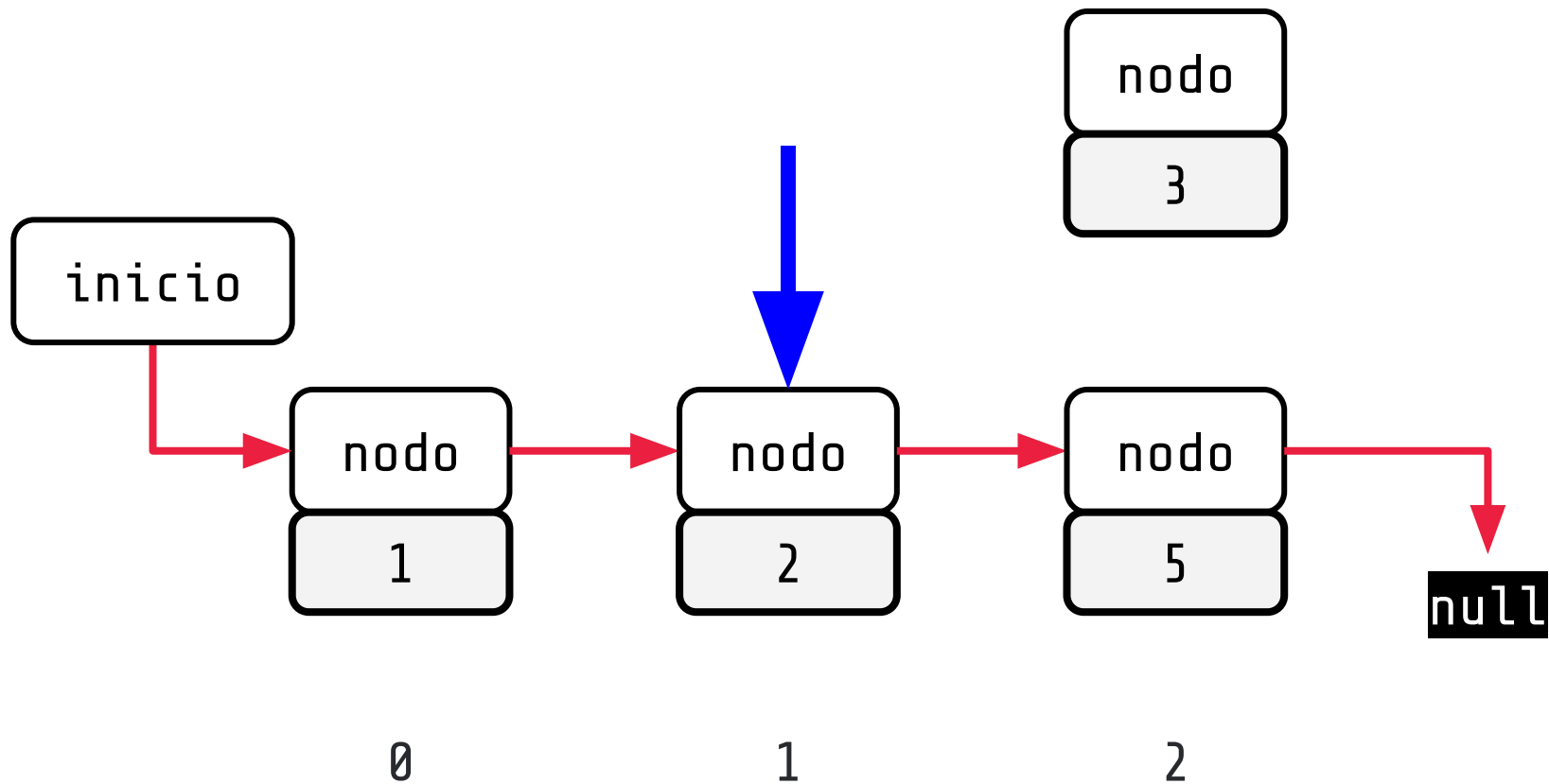


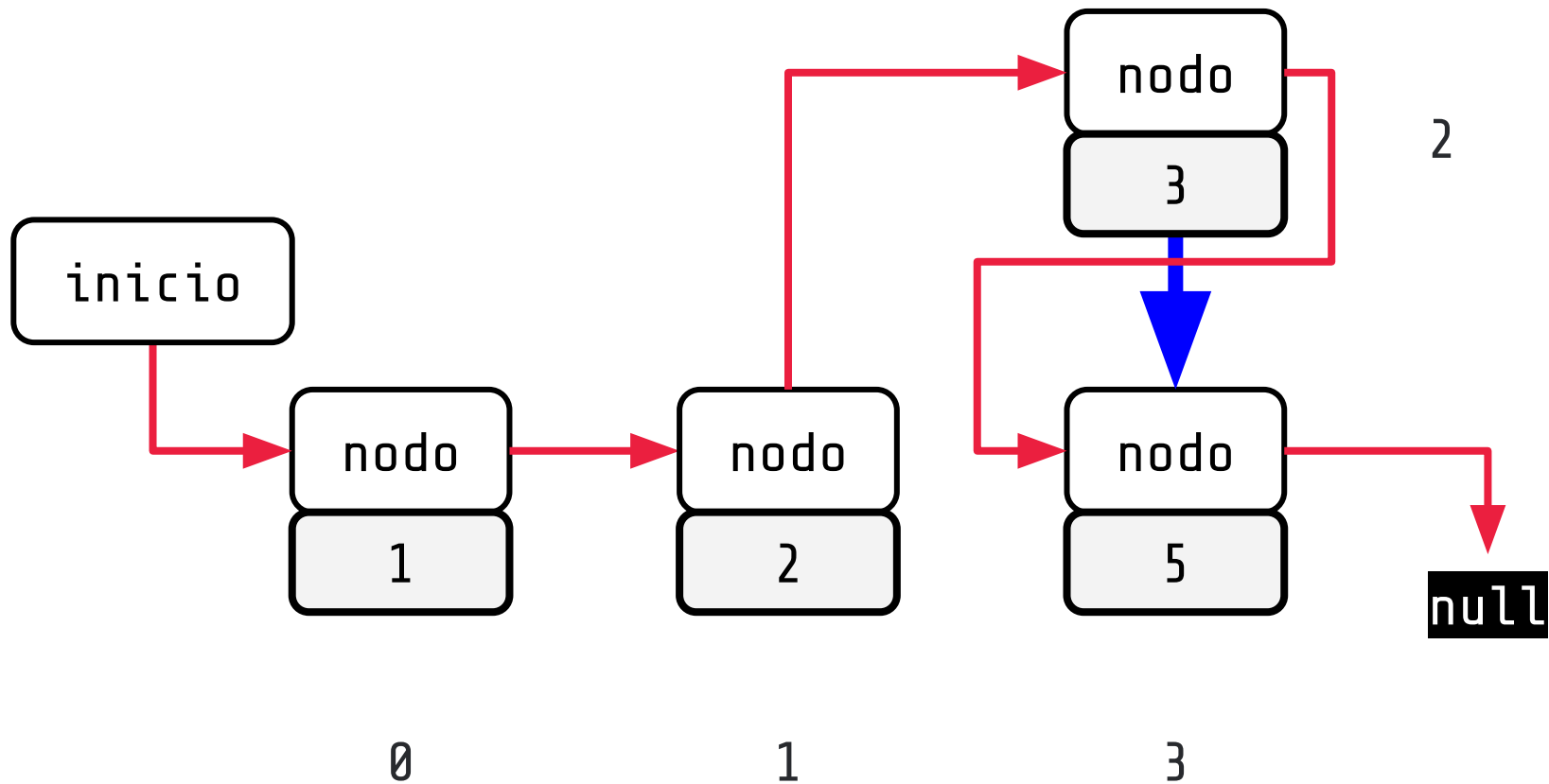
**Pero mover  
elementos es más  
eficiente**











# Las clases involucradas

```
public class ListaEnlazadaInt{  
    private Nodo inicio;  
    public void insertar(int posicion, int valor);  
    public int largo();  
    public int extraer(int posicion);  
    public void modificar(int posicion, int valor);  
}
```

```
public class NodoInt{  
    private NodoInt siguiente;  
    private int valor;  
    public NodoInt siguiente();  
    public boolean hayMas();  
}
```



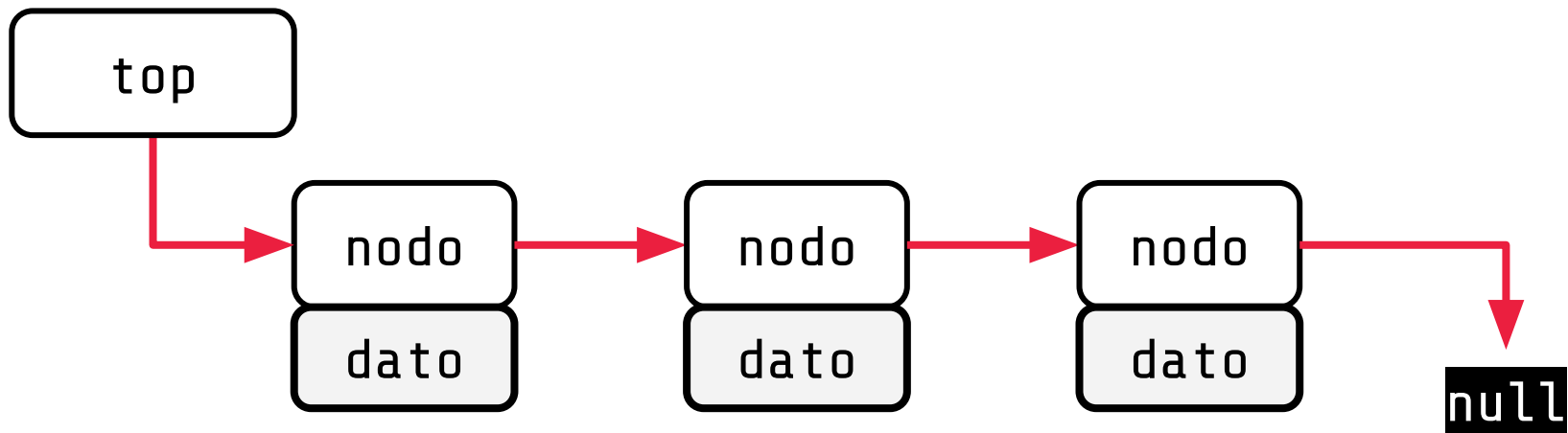
**¿Preguntas?**

---

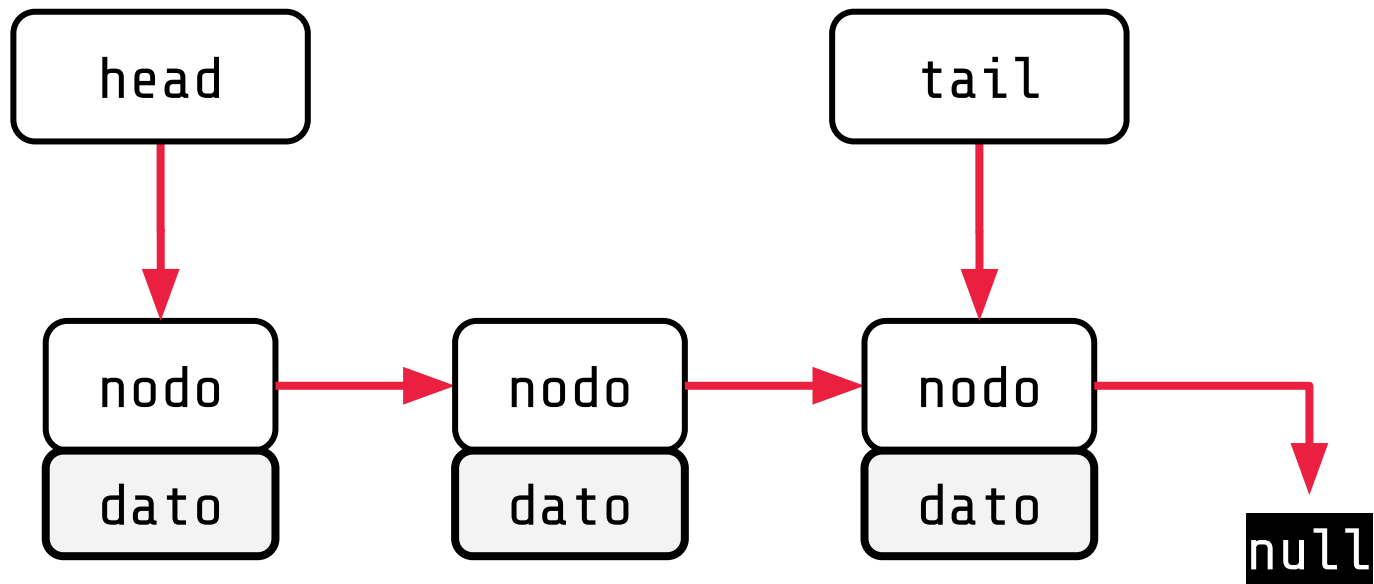
# Son la base de otras estructuras

# Pila (Stack)

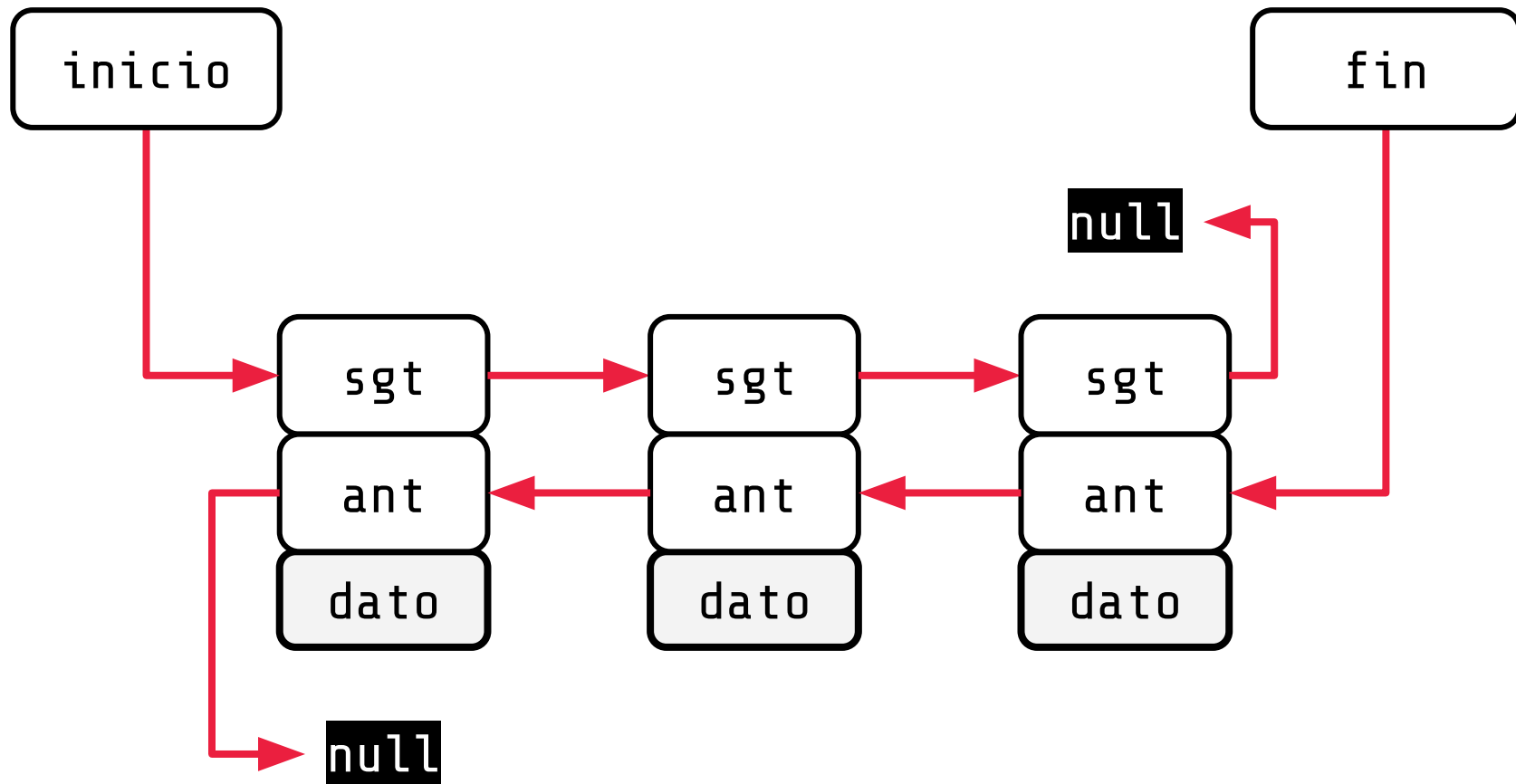




# Cola (queue)



# Listas doblemente enlazadas

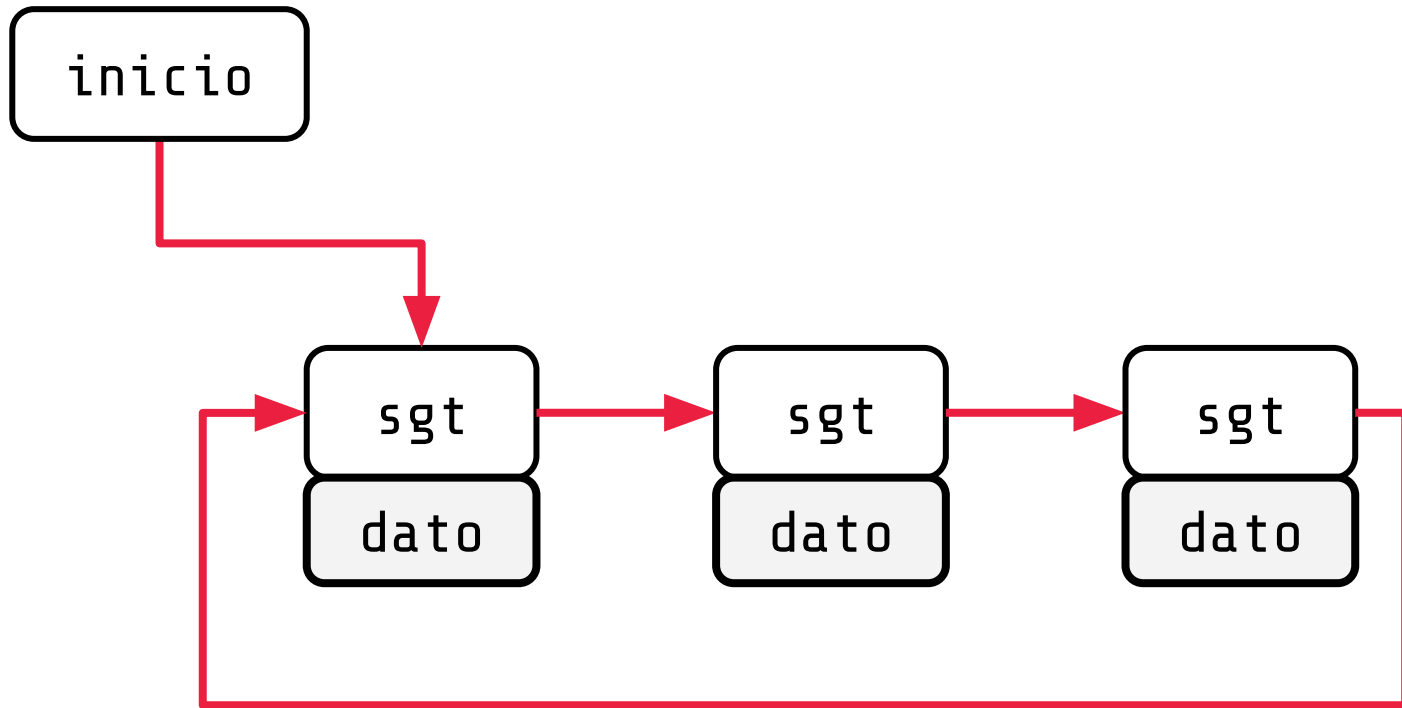


---

# Usos

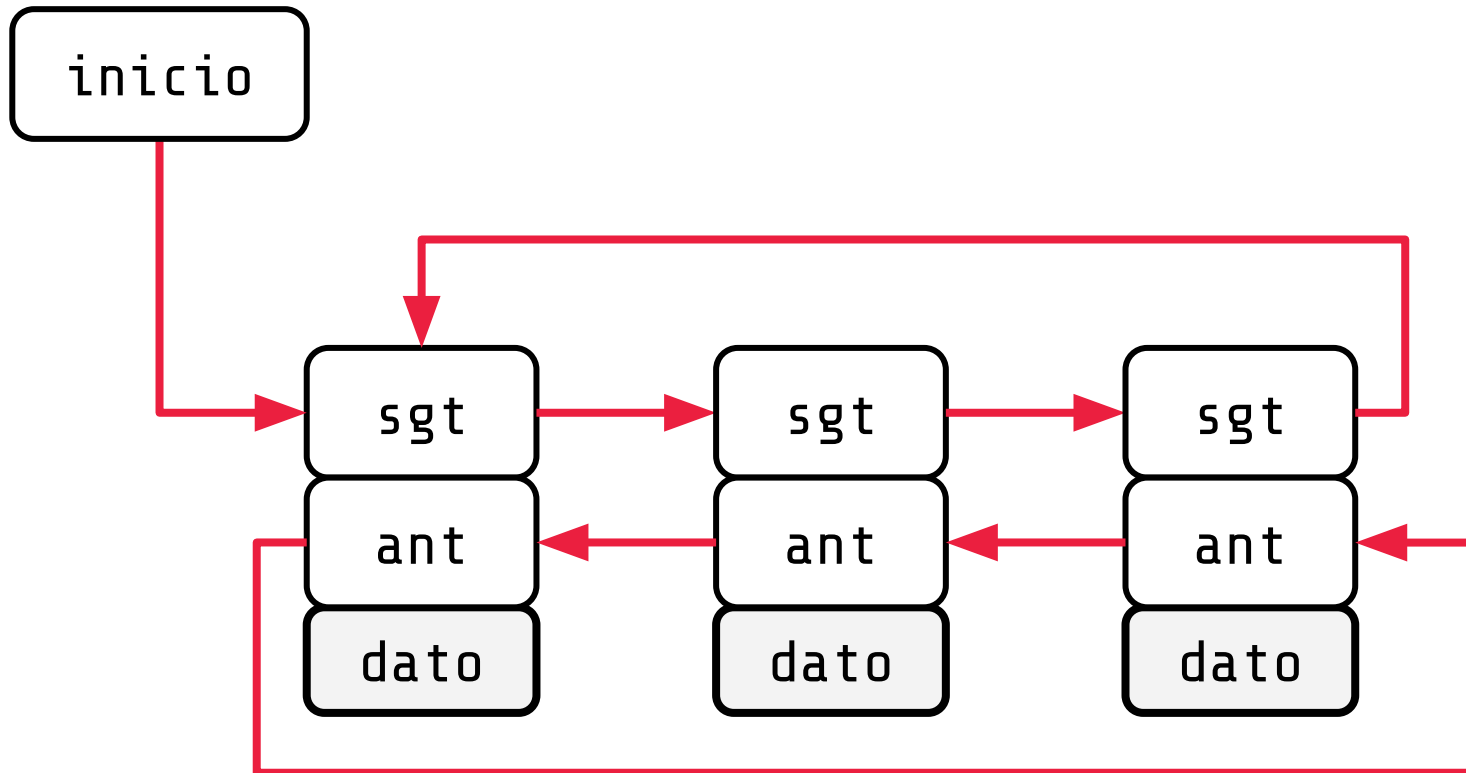
1. Inserción y eliminación eficiente, solo es necesario 'mover' un par de referencias.
2. Recorrido bidireccional
3. Queremos implementar un Arreglo que pueda crecer
4. Gestionar memoria
5. Implementar otras estructuras de datos avanzadas

# Lista enlazada circular





# Lista enlazada doble circular





**¿Preguntas?**

**unrn.edu.ar**

**UNRN**

Universidad Nacional  
de **Río Negro**



| **unrionegro**