TAD Lista

ALGORITMOS Y ESTRUCTURAS DE DATOS I

Tipo Abstracto de Datos (TAD)

- Un TAD es **un tipo de datos junto con las operaciones definidas para él (procedimientos de acceso)**, independientemente de su implementación.

- Si bien permite una especificación con precisión, no está dado como un tipo de datos concreto del lenguaje.

Tipo Abstracto de Datos (TAD)

Como se define?

Básicamente dándole un nombre y asociando a él un conjunto de operaciones aplicables a los elementos del tipo. **Definimos qué es lo que hace y no cómo lo hace.**

Como se implementa?

Asociando un método (código) a cada una de las operaciones definidas sobre el conjunto. **Definimos cómo lo hace.**

Tipo Abstracto de Datos (TAD)

Ventajas

- La separación de la especificación (definición) y la implementación ayuda a reducir la complejidad de la tarea a realizar.

- El cambio en la implementación de alguna de las operaciones del TAD no afecta a los programas que utilizan las operaciones de dicho TAD.

TAD Lista

- Una lista es una estructura de datos que contiene una secuencia lineal de un número arbitrario de ítems del mismo tipo.

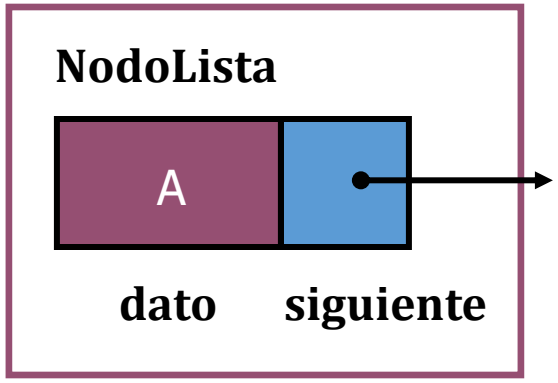
- Llamaremos Nodo a cada elemento de la lista. Cada nodo tiene al menos:

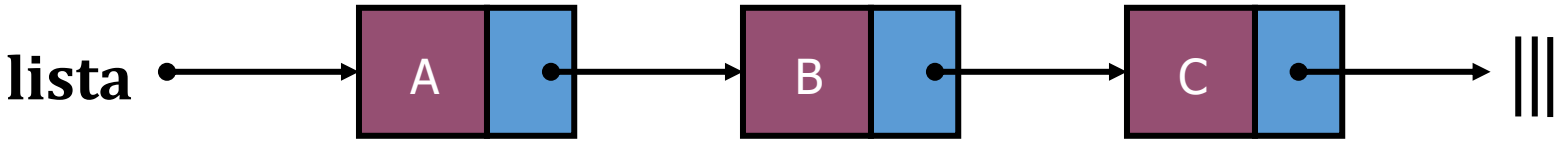
✓ Un ítem de información (de cualquier tipo).
✓ Un puntero al siguiente nodo de la lista.

TAD Lista (posibles operaciones)

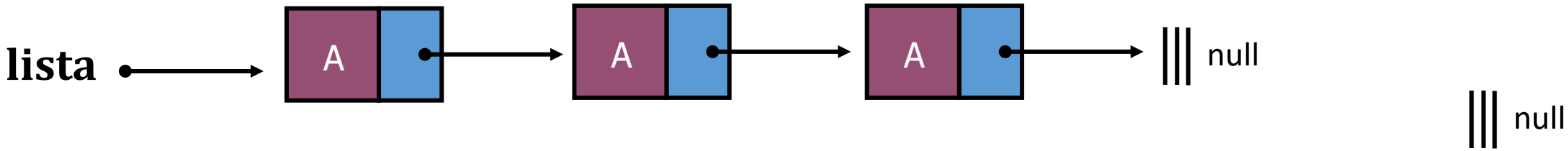
```
public boolean esVacia();
public void agregarInicio(int n);
public void agregarFinal(int n);
public void borrarInicio();
public void borrarFin();
public void vaciar();
public void mostrar();

public void agregarOrd(int n);
public void borrarElemento(int n);
public int cantElementos();
public NodoLista obtenerElemento(int n);
public void mostrarREC(NodoLista l);
```

Lista Simplemente Encadenada



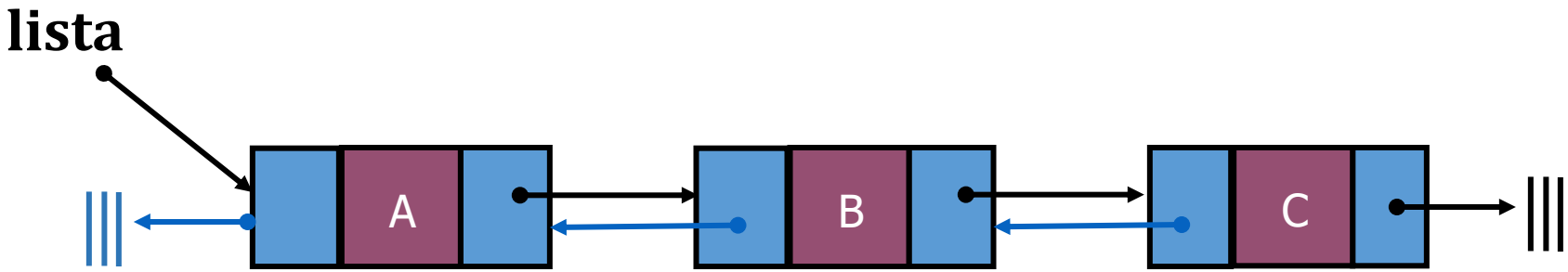


Lista Simplemente Encadenada



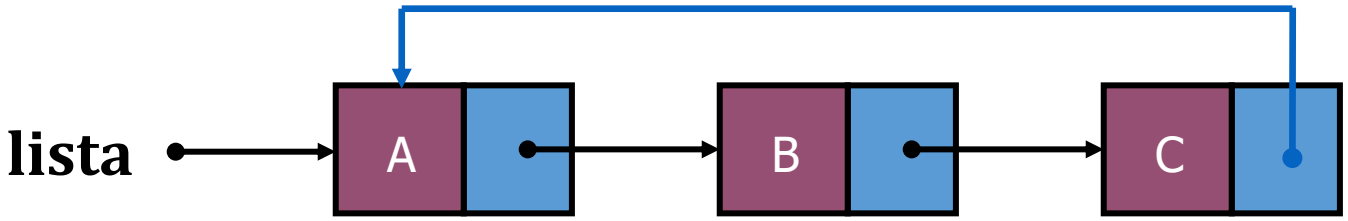
Lista Doblemente Encadenada

Para facilitar el recorrido de una lista en cualquier dirección, podemos agregar un apuntador adicional a cada nodo, que apunte al nodo anterior de la lista.



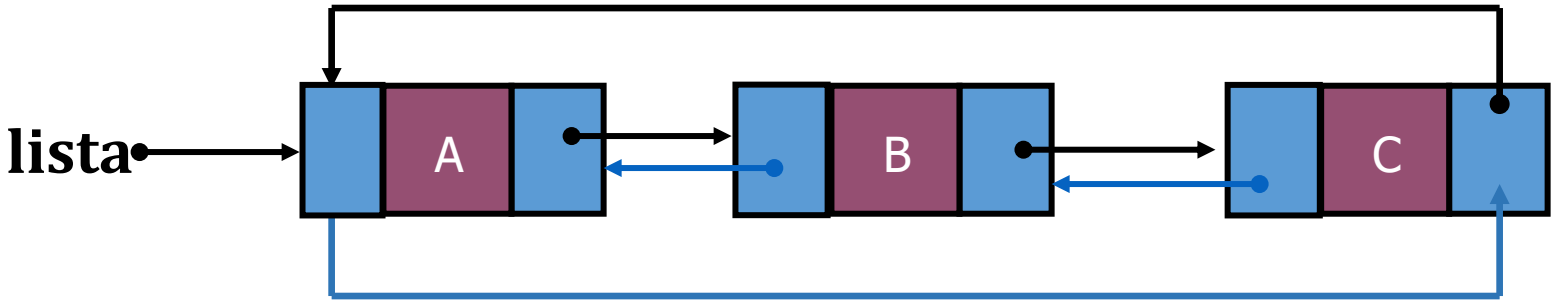
Lista Circular

Para recorrer la lista precisamos mantener una referencia (puntero) al primer elemento para saber cuando debemos parar la recorrida. No hay punteros a NULL



Lista Circular doblemente encadenada

Es la unión de una lista doblemente encadenada con una lista circular



TAD Lista

Ventajas frente a otras opciones

Al definir listas estamos trabajando con estructuras dinámicas, por lo cual no es necesario especificarle su tamaño en tiempo de compilación. Una lista puede crecer y contraerse en tiempo de ejecución.

Es más eficiente al realizar ciertas operaciones, respecto a otras opciones, como por ejemplo vectores.