



# Estructura de Datos y Algoritmos

ITBA 2024-Q2

## TP 3C- Ejer 3

Implementar el iterador read-only.

Agregar a la interface:

```
public interface  
SortedListService<T extends  
Comparable<? super T>>  
extends Iterable<T>
```

Caso de Uso:

```
SortedListService<String> l =  
new SortedLinkedList<>();  
l.add("hola"); l.add("tal"); l.add("que");  
for (String s : l) {  
    System.out.println(s);  
}
```



Iterable<T>

hay que implementar:

**public Iterator<T> iterator()**



Iterator<T>

hay que implementar:

**public boolean hasNext()**

**public T next()**

Caso de Uso From Scratch:

```
SortedListService<String> l =
```

```
new SortedLinkedList<>();
```

```
l.add("hola"); l.add("tal"); l.add("que");
```

```
for( Iterator<String> it = l.iterator(); it.hasNext(); ) {
```

```
    System.out.println( it.next() );
```

```
}
```

## Posible solución:

@Override

```
public Iterator<T> iterator() {  
    return new SortedLinkedListIterator() {  
    };  
}
```



```
private class SortedLinkedListIterator implements Iterator<T> {  
    private Node current;
```

```
    public SortedLinkedListIterator() {  
        current= root; }  

```

```
    @Override  
    public boolean hasNext() {  
        return current != null; }  

```


```
    @Override  
    public T next() {  
        if (!hasNext()) {  
            throw new NoSuchElementException();  
        }  
        T rta= current.data;  
        current= current.next;  
        return rta;  
    }  
}
```



# TP 3C- Ejer 4

Que complejidad temporal  
tienen:

**getMin()**  
**getMax()**  
**size()**



Algunas de esas operaciones podría bajar su  
complejidad, a expensas de un poco más de  
información



Sería interesante evitar “recorrer la lista” para saber su tamaño.

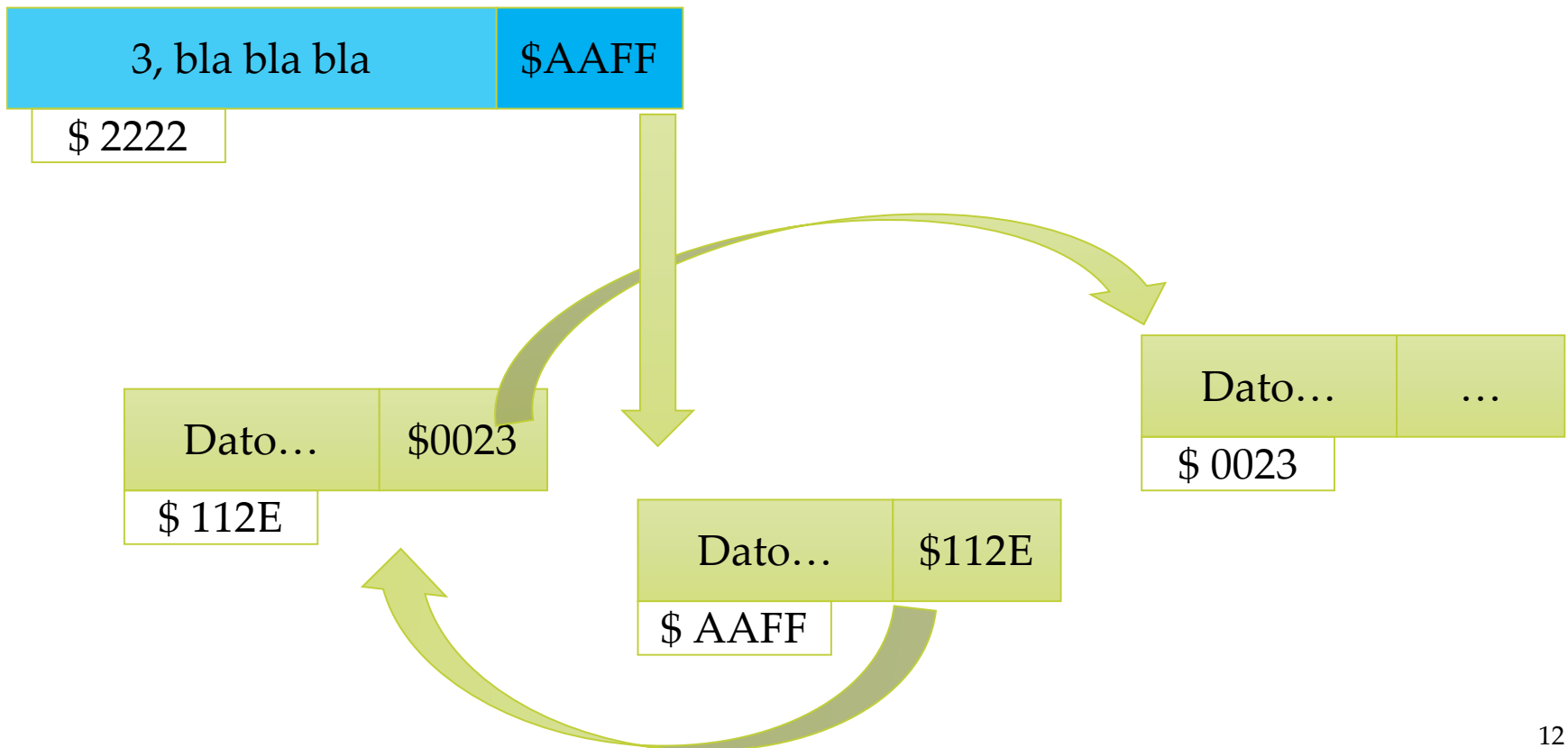
Guardarlo en informacion “global” a la lista. .

Analicemos otra variante de Lista Lineal Simplemente Encadenada.

# Analizando otras alternativas

Alternativa: “**Lista lineal simplemente encadenada con header**”

Header: \$2222”



## Definición Lista Lineal Simplemente Encadenada con Header

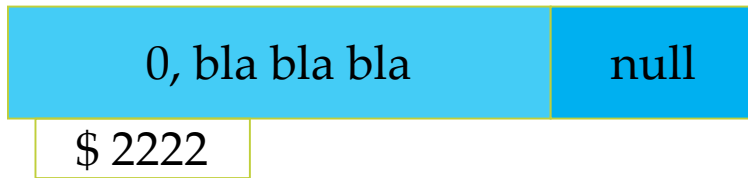
Es una estructura de datos compuesta por:

- Un elemento distinguido llamado “header” que tiene la referencia del primer elemento de la lista **y además información global de la lista.**
- Cada nodo/elemento (común) almacena 2 cosas: su info y la referencia al elemento siguiente.

# Analizando otras alternativas

Alternativa: “**Lista lineal simplemente encadenada con header**”

Header: \$2222”



## Variante para Índice

### Definición Lista Lineal Simplemente Encadenada **Ordenada** con Header

Es una lista lineal simplemente encadenada con header que además mantiene los **elementos ordenados** con algún criterio de ordenación.



Es decir,

- Hay 2 tipos de nodos: header y comunes.
- El nodo header no tiene que ser comparable. Hay uno solo de ese tipo de nodo!!!
- Los nodos comunes tienen que poder compararse entre sí.



## TP 3C- Ejer 5

Implementar la clase  
**SortedLinkedListWithHeader**  
(implementa misma interface)  
donde:

**getMin(), getMax(), size()**

**Resuelvan en  $O(1)$**