

## Laboratorio Nro. 3: Listas enlazadas y listas hechas con arreglos

**Manuel García Jimenez**  
Universidad Eafit  
Medellín, Colombia  
megarciaj@eafit.edu.co

**Isaias Labrador Sanchez**  
Universidad Eafit  
Medellín, Colombia  
ilabradors@eafit.edu.co

### 1) Documentación Git:

- **unoConArrayList:**

Este metodo multiplica todos los números en una lista de enteros.

@param: a tipo ArrayList<Integer>

Lista de enteros

@return: Int mult

La multiplicación de todos los elemntos de la Lista.

- **unoConLinkedList:**

Este metodo multiplica todos los números en una lista de enteros.

@param: a tipo LinkedList<Integer>

Lista de enteros

@return: Int mult

La multiplicación de todos los elemntos de la Lista.

- **SmartInsertArrayList:**

Este metodo inserta un element al final de una lista si solo si este elemto no esta ya en la lista

@param: a tipo ArrayList<Integer>

Lista de enteros

@param: data tipo entero

El número a insertar

- **SmartInsertLinkedList:**

Este metodo inserta un element al final de una lista si solo si este elemto no esta ya en la lista

@param: a tipo LinkedList<Integer>

Lista de enteros

@param: data tipo entero

El número a insertar

- **tresConArrayList:**

Este metodo divide una lista de enteros tal que ambos lados sumen al rededor de lo mismo.

@param: a tipo ArrayList<Integer>

Lista de enteros a dividir

@return: Int

La posición la cual al dividir la lista en los que estan a la derecha y los que estan a la izquierda, hace que ambos lados sumen más o menos lo mismo

•tresConLinkedList:

Este metodo divide una lista de enteros tal que ambos lados sumen al rededor de lo mismo.

@param: a tipo LinkedList<Integer>

Lista de enteros a dividir

@return: Int

La posición la cual al dividir la lista en los que estan a la derecha y los que estan a la izquierda, hace que ambos lados sumen más o menos lo mismo

•ejercicio4:

Este metodo nos dice que neveras se lleva cada tienda que le solicita neveras a la bodega.

@param: neveras tipo LinkedList<Integer>

Lista de enteros que representan las neveras en el orden en que fueron metidas a la bodega.

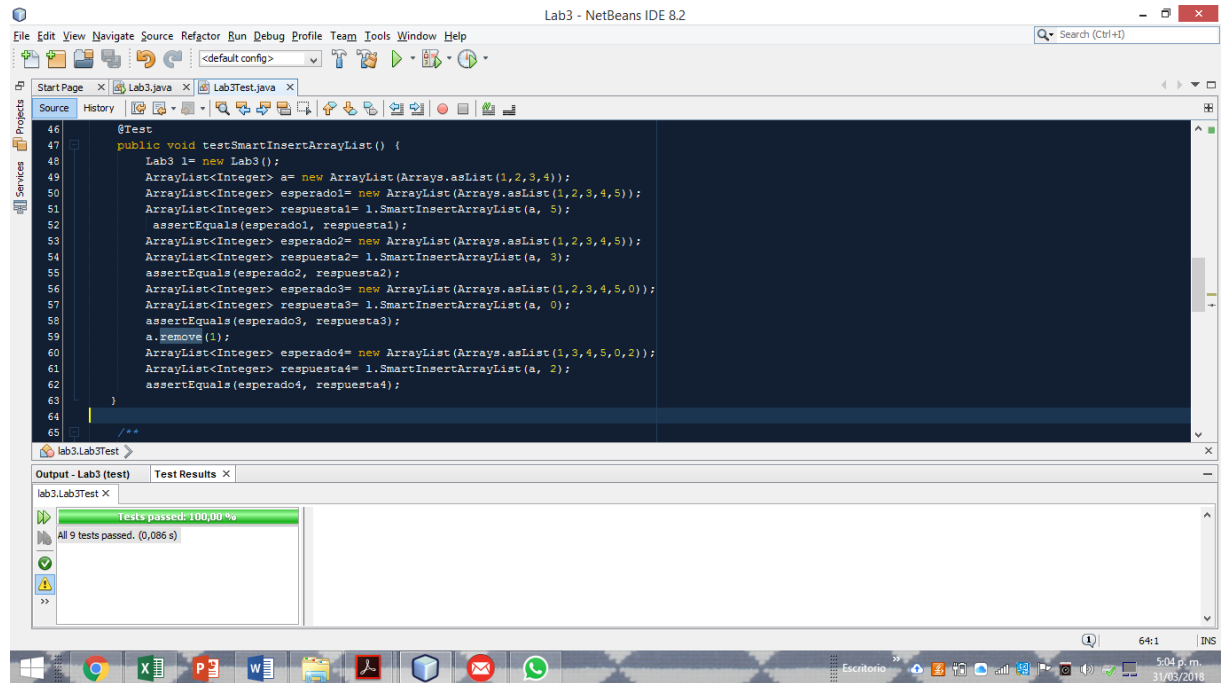
@param: solicitudes tipo LinkedList<Integer>

Lista de enteros que representa el número de neveras que pidio cada

### 3) Simulacro de preguntas de sustentación de Proyectos

1.

	ArrayList	LinkedList
Ejercicio 1	$O(n)$	$O(n^2)$
Ejercicio 2	$O(n)$	$O(n^2)$
Ejercicio 3	$O(n)$	$O(n^2)$
Ejercicio 4	$O(n^2)$	$O(n^2)$



2.

3.

### **Implementación ejercicio 2.2:**

Primero se pide al usuario ingresar una línea de texto con “fallos” (Líneas que contienen saltos por la tecla inicio y tecla fin), después se crean 3 listas simplemente enlazadas que contengan: las cadenas de caracteres que van al inicio, las cadenas de caracteres que van al final, y aquellas que van en la “mitad” es decir las que no están afectadas por algún ‘[’ o ‘]’. Después se procede a recorrer la cadena de texto ingresada por el usuario letra por letra, teniendo como auxiliar una cadena de texto vacía que va añadiendo cada carácter diferente de ‘[’ o ‘]’ a sí misma y un entero que me dice si hemos pasado algún ‘[’ (1), algún ‘]’ (2) o si la cadena no está afectada por ninguno de los dos (0). Si al momento de recorrer la cadena se llega a un carácter ‘[’ se toma la cadena auxiliar con los caracteres hasta ese punto y se añade en la posición 0 de la lista “inicio”, se vuelve a dejar la cadena auxiliar vacía y se sigue llenando con los caracteres que siguen, lo mismo en caso de encontrarse con el carácter ‘]’ con la diferencia de almacenarlo en “final” en la última posición, al finalizar el proceso de recorrido de la cadena ingresada por el usuario, se concatenan las cadenas de inicio, las de mitad, y las de final, en ese orden y se obtiene la cadena corregida.

4. Seria  $T(N * M)$

5. La N representa la longitud de la cadena de caracteres que el usuario ingresa y M representa el número de “errores” que la cadena contenga (‘[’ o ‘]’ o ninguno(1))

### **4) Simulacro de Parcial**

1. A
2. B
3.  $Q.size() < 0$   
 $<=$   
 $q.getLast()$

q.remove()

4. lista.size()  
lista.addFirst(auxiliar.pop());
5. auxiliar1.size()<0 auxiliar2.size()<0  
persona.offer(edad)
6. C