	<b>UNIVERSIDAD EAFIT</b> <b>ESCUELA DE INGENIERÍA</b> <b>DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</b>	Código: ST245
		Estructura de Datos 1

## Laboratorio Nro. 4: Implementación de listas enlazadas

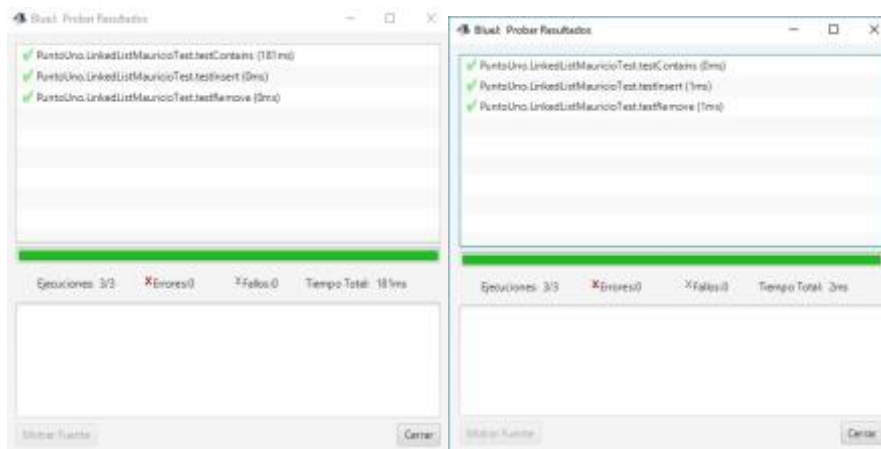
**Alejandra Ossa Yepes**  
**1**  
**Universidad Eafit**  
**Medellín, Colombia**  
**aossay@eafit.edu.co**

**Jose Andres Carvajal**  
**2**  
**Universidad Eafit**  
**Medellín, Colombia**  
**jacarvajab@eafit.edu.co**

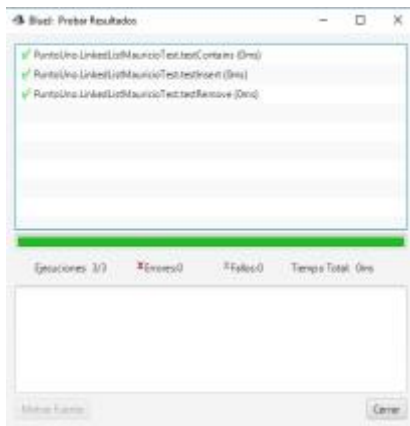
**Brian Fernando Morales**  
**3**  
**Universidad Eafit**  
**Medellín, Colombia**  
**bfmoralesa@eafit.edu.co**

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

1. Verifiquen, utilizando JUnit, que todos los tests escritos en el numeral 1.2 pasan. Muestren, en su informe de PDF, que los tests se pasan correctamente; por ejemplo, incluyendo una imagen de los resultados de los tests.



**DOCENTE MAURICIO TORO BERMÚDEZ**  
**Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627**  
**Correo: mtorobe@eafit.edu.co**



2. Expliquen con sus propias palabras cómo funciona el ejercicio en línea del numeral 2.1

El algoritmo está basado en ordenes que se hacen dentro de la consola, las cuales generan el cambio de posición de unos bloques que están enumerados y ordenados en su creación; el número de bloques se encuentran finitos y también las ordenes están predisuestas en el diseño del algoritmo.

En el método main() se encuentran los lectores de la entrada, la recolección de esta información se hace por medio de `BufferedReader`, el cual nos permite registrar la entrada del usuario en modo de consola, este algoritmo trabaja con pilas que nos facilitan el manejo y eficiente de los bloques, al principio se encuentra un ciclo que llena nuestra estructura de bloques de acuerdo al tamaño que el usuario indique, después se hace un ciclo que registra la entrada del usuario y separa la cadena de entrada por medio de un `StringTokenizer` token, que es el que recolectar la información por partes y así hacer un manejo individual de la información registrada, la primera cadena es el movimiento que vaya a hacer el usuario ya se move o pile, cada una de estas opciones entra por medio de una condición, la cual hace un llamado al método que genera este movimiento, en el caso de move puede ser (`moveOnto`, `moveOver`) dependiendo de la segunda entrada del usuario,

y en el caso de pile también está la opción de (pileOnto, pileOver) las cuales también tienen sus propios métodos, además se registran el número de bloque con la posición en la que se va a cambiar este.

Estos métodos hacen el reordenamiento de los bloques con respecto a la entrada del usuario y distribuyen cada bloque al lugar indicado además de que estos movimientos están bajo las reglas del juego que son las que hacen el comportamiento de este. Después de hacer todo este proceso se genera una salida que registra el estado final de la pila, la cual muestra en qué posición quedó finalmente cada bloque.

3. Calculen la complejidad del ejercicio en línea del numeral 2.1 y agréguenla al informe PDF

complejidad  $O(n)$

Este algoritmo cuenta con muy poca complejidad asintótica, debido a que tiene muy pocos ciclos de recorrido y una capacidad de llamados metódicos muy efectiva y creativa a la hora de la ejecución, además hace muy buen manejo de las opciones de Java para las funciones que necesitan.

4. Expliquen con sus palabras las variables (qué es 'n', qué es 'm', etc.) del cálculo de complejidad del numeral 3.4

En la complejidad del algoritmo  $n$  representa el número de veces que se repite una instrucción y  $c$  los tiempos que son constantes en el programa.


#### 4) Simulacro de Parcial

1. 03. lista.size(),  
07. lista.add(auxiliar.pop());
2. 12. auxiliar1.size()>0  
16. auxiliar2.size()>0  
18. persona.offer(edad);
3. c)  $O(n^2)$

#### 5) Referencias de los códigos

Title: 101 - The Blocks Problem.java

Author: AhmedHani

	<p style="text-align: center;">UNIVERSIDAD EAFIT ESCUELA DE INGENIERÍA DEPARTAMENTO DE INFORMÁTICA Y SISTEMAS</p>	<p>Código: ST245</p> <p>Estructura de Datos 1</p>
---	---	---

Date: 11 Jul 2013

Code versión: 0.1

Availability:

<https://github.com/AhmedHani/Online-Judges-Problems-SourceCode/blob/master/UVa/101%20-%20The%20Blocks%20Problem.java>

<http://codigolibre.weebly.com/blog/listas-simples-en-java>

- a. El reporte de cambios del informe de laboratorio