

## Laboratorio Nro. 3: Utilización de Listas Enlazadas (*Linked List*) y Listas Hechas con Arreglos (*Array List*)

**Kevin Arley Parra Henao**  
Universidad EAFIT  
Medellín, Colombia  
kaparrah@eafit.edu.co

**Daniel Alejandro Mesa Arango**  
Universidad EAFIT  
Medellín, Colombia  
damesaa@eafit.edu.co

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

Indicados en los respectivos numerales.

#### 3.1) Complejidades de los numerales del punto 1.

	LinkedList	ArrayList
Ejercicio 1.1	Sin iterador $O(n^2)$ Con iterador $O(n)$	$O(n)$
Ejercicio 1.2	$O(1)$ al final(peor de los casos)	$O(n)$ al final(peor de los casos)
Ejercicio 1.3	$O(n^2)$	$O(n)$
Ejercicio 1.4	$O(n*m)$	$O(n*m)$

#### 3.2) Explicación del ejercicio 2.1.

**Importamos las librerías**

**Librerías para las lecturas de datos**

```
import java.io.BufferedReader;  
import java.io.InputStreamReader;  
import java.io.IOException;
```

**Librería para las listas enlazadas**

```
import java.util.LinkedList;
```

```
public class ejercicio21 {
```

DOCENTE MAURICIO TORO BERMÚDEZ

Teléfono: (+57) (4) 261 95 00 Ext. 9473. Oficina: 19 - 627

Correo: [mtorobe@eafit.edu.co](mailto:mtorobe@eafit.edu.co)

**Declaramos y definimos la lista enlazada que vamos a usar para procesar los caracteres ingresados**

```
LinkedList<String> teclado = new LinkedList<String>();
```

### **Método principal**

```
public static void main(String[] args) throws IOException {
```

**Declaramos y definimos el Buffer que usaremos para leer la cadena de caracteres**

```
    BufferedReader br = new BufferedReader(new  
    InputStreamReader(System.in));
```

**Almacenamos la cadena de caracteres ingresada mediante el método readLine**

```
    String entrada = br.readLine();
```

**Instanciamos un objeto de la clase que estamos programando para poder usar los métodos programados**

```
    ejercicio21 ej = new ejercicio21();
```

**Procesamos la cadena con el método teclear y almacenamos la salida en un string que después imprimimos**

```
    String out = ej.teclear(entrada).toString();
```

```
    System.out.println(ej.imprimir(out));
```

```
}
```

**Método que se usa para procesar la cadena de texto leída**

```
public LinkedList<String> teclear(String entrada) {
```

**Variable que utilizaremos para el control del almacenaje en la lista enlazada**

```
    int j = 0;
```

**Recorremos la cadena de caracteres ingresada**

```
    for (int i = 0; i < entrada.length(); ++i) {
```

**Si en la cadena de caracteres hay una corchete de apertura, significa que se presionó la tecla inicio y se debe ir a la posición inicial de la lista enlazada, es decir, j es igual a cero.**

```
        if (entrada.charAt(i) == '[') {
```

```
            j = 0;
```

```
        } else if (entrada.charAt(i) == ']') {
```

**Si en la cadena de caracteres hay una corchete de cierre, significa que se presionó la tecla fin y se debe ir a la posición final de la lista enlazada, es decir, j es igual al tamaño de la lista enlazada(que representa la última posición).**

```
            j = teclado.size();
```

```
        } else {
```

**En otro caso, se agrega el carácter en la posición i de la cadena de caracteres a la lista enlazada en la posición j y se aumenta en una posición la variable que**

estamos usando para el control del almacena en la lista enlazada, es decir la variable j.

```
        tecleado.add(j, entrada.charAt(i)+"");  
        j++;  
    }  
}
```

**Retornamos la lista enlazada con la entrada modificada.**

```
    return tecleado;  
}
```

**Método para hacer más clara la visualización del resultado, suprimiendo las comas que agregan por defecto en el método toString de la clase LinkedList.**

```
public String imprimir(String str ){  
    String r = "";  
    for(int i = 0; i < str.length(); ++i){  
        if(!(str.charAt(i) == ',' )){  
            r = r + str.charAt(i);  
        }  
    }  
    return r;  
}  
  
}
```

### 3.3) complejidad del ejercicio 2.1.

**Se calcula la complejidad del método teclear ya que esa sería la respuesta general al ejercicio**

```
public LinkedList<String> teclear(String entrada) {  
    int j = 0; //C1  
    for (int i = 0; i < entrada.length(); ++i) { //C2 + C3 *n  
        if (entrada.charAt(i) == '[') { //C4 * n  
            j = 0;  
        } else if (entrada.charAt(i) == ']') {  
            j = tecleado.size();  
        } else {  
            tecleado.add(j, entrada.charAt(i)+""); //C5*n  
            j++;  
        }  
    }  
}
```

```
return teclado;  
}
```

**Complejidad:**

$$T(n) = C1 + C2 + C3*n + C4*n + C5*n$$

$$T(n) = C + C*n$$

luego  $T(n)$  es  $O(n)$

**3.4) Explicación de las variables del cálculo de complejidad en el punto 3.3.**

La variable  $n$  en el cálculo de la complejidad del ejercicio 2.1 se refiere al tamaño de la cadena de caracteres ingresada.

**4) Simulacro de Parcial**

1. a
2. c
3. A continuación.
  - A. `!(q.size() == 1)`
  - B. `<=`
  - C. `q.pop()`
  - D. `q.getLast();`

## 6. Trabajo en Equipo y Progreso Gradual (Opcional)

### a) Actas de reunión

Integrante	Fecha	Hecho	Haciendo	Por hacer
Kevin Parra	23/09/2017	Punto 2.1	Punto 3, crear informe en drive	Subir código e informe
Daniel Mesa	24/09/2017		Punto 1 e items	tabla de complejidades
Kevin Parra	24/09/2017	Punto 3	Punto 4	Poner acta en el informe
Daniel Mesa	24/09/2017	Punto 1	Punto 3.1	punto 3 del 4
Daniel Mesa	24/09/2017		Revisar	Entregar

**b) El reporte de cambios en el código**

commit 531bb412c0c0e33c92e76cd68046862e08027c4c (HEAD -> master, origin/master, origin/HEAD)

Author: Daniel Mesa <damesaa@eafit.edu.co>

Date: Sun Sep 24 23:17:24 2017 -0500

terminado 1 lab y 3.1

commit 21b6d862ce5ff4730b20e68de35a3dd4ba11f236

Author: Daniel Mesa <damesaa@eafit.edu.co>

Date: Sun Sep 24 22:20:36 2017 -0500

casiterminado el lab

commit 4abb38d631a1cc6db0d6c08cacc99462380dfa96

Author: eafit-201710093010 <30359351+eafit-

201710093010@users.noreply.github.com>  
Date: Sat Sep 23 23:38:39 2017 -0500

solo falta el punto tres


Mañana lo hago

commit b4ef49041a626c878931eea6c7e42c488f2e1289

Author: Daniel Mesa <damesaa@eafit.edu.co>

Date: Sat Sep 23 21:16:09 2017 -0500

Agregando estructura

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

c) El reporte de cambios del informe de laboratorio

Historial de versiones

Mostrar solo las versiones con nombre
☐

Hoy

24 de septiembre, 23:13

⋮

Versión actual

Daniel Mesa

24 de septiembre, 22:00

Daniel Mesa

▶ 24 de septiembre, 18:15

kevin parra

▶ 24 de septiembre, 17:18

kevin parra

▶ 24 de septiembre, 0:47

Daniel Mesa

Ayer

▶ 23 de septiembre, 23:40

kevin parra

23 de septiembre, 23:39

kevin parra

a. El reporte de cambios del informe de laboratorio

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