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

## Laboratorio Nro. 3: Listas

<b>Kevyn Santiago Gómez Patiño</b> Universidad Eafit Medellín, Colombia ksgomezp@eafit.edu.co	<b>Luis Javier palacio</b> Universidad Eafit Medellín, Colombia ljpalaciom@eafit.edu.co
<b>Santiago castrillon</b> Universidad Eafit Medellín, Colombia scastrillg@eafit.edu.co	

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

1.

	ArrayList	LinkedList
Ejercicio 1.1	$O(n)$	$O(n^2)$
Ejercicio 1.2	$O(n^2)$	$O(n)$
Ejercicio 1.3	$O(n^2)$	$O(n^3)$
Ejercicio 1.4	$O(n^2)$	$O(n^3)$

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)

2. Tenemos una lista de caracteres y usamos la clase Scanner para guardar lo que se ingresa en una cadena de caracteres. Tenemos una variable booleana inicio que se activará cuando se encuentre un '[' y se desactivará si encuentra un ']'. También tenemos una variable entera index que será la posición donde iremos añadiendo los caracteres mientras el booleano inicio sea verdadero. Hacemos un ciclo que recorre la cadena de caracteres y usamos 'switch' para todas las posibilidades. En un primer caso, si el carácter actual es '[' activamos (dar un valor verdadero) al booleano 'inicio'. En un segundo caso si el carácter actual es ']' entonces desactivamos el booleano inicio y normalizamos el index en 0 (esto para que cada vez que activemos el booleano agregue desde el principio). En el caso "default" si la variable inicio está desactivada, añadimos el carácter al final de la lista y si inicio está activado entonces añadimos el carácter en la posición que indica la variable "index". Al final, con un ciclo imprimimos cada carácter al tiempo que lo eliminamos de la lista, y cuando esta esté vacía el ciclo termina y el método termina su ejecución.

3. **Su complejidad es de  $O(n)$**

```
public static void linea() {  
    LinkedList<Character> list = new LinkedList<>(); C1  
    Scanner sc = new Scanner(System.in); //C2  
    String str = sc.nextLine(); //C3  
    boolean inicio = false; //C4  
    int index = 0; //C5  
    for (int i = 0; i < str.length(); i++) {  
        switch (str.charAt(i)) { C6 * n  
            case '[': C7*n  
                inicio = true; C8 *n  
                break;  
            case ']':  
                inicio = false; C8*n  
                index = 0;  
        }  
    }  
}
```

```
        break;
    default:
        if (!inicio) { C10*n
            list.add(str.charAt(i)); C11*n
        } else {
            list.add(index++, str.charAt(i)); C12*n
        } break;
    }
}
```

```
while (!list.isEmpty()) {
    System.out.print(list.remove()); C13*n
}
```

- $T(n) = C + C' * n$
- $T(n)$  es  $O(C + C' * n)$
- $T(n)$  es  $O(C' * n)$
- $T(n)$  es  $O(n)$


#### 4.

- *String str* : Guarda la línea ingresada por el Scanner
- *Int i* Es un iterador que nos sirve para movernos dentro de los ciclos
- *Boolean inicio*: Es una variable que sirve para simular el hecho que se haya oprimido la tecla inicio en un teclado.
- *Int index*: Es un entero que nos sirve para determinar la posición dónde iremos añadiendo los caracteres mientras el booleano inicio sea verdadero.

#### 4) Simulacro de Parcial

1. a
2. c
3. 

```
while (q.size() > 1){ for (int i = 1; i <= num; i++) q.add(q.remove());
q.remove();
}
return q.remove();
```

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