

## Laboratorio Nro. 3

### Listas y Vectores Dinámicos

**Julián Gómez Benítez**  
Universidad Eafit  
Medellín, Colombia  
jgomezb11@eafit.edu.co

**Juan Pablo Rincon Usma**  
Universidad Eafit  
Medellín, Colombia  
jprinconu@eafit.edu.co

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

#### 3.1

	<b>ArrayList o Vector</b>	<b>LinkedList</b>
<b>Ejercicio 1</b>	<b><math>O(n)</math> en el algoritmo, <math>O(n)</math> para insertar un elemento</b>	<b><math>O(n)</math> en el algoritmo, <math>O(1)</math> para insertar un elemento</b>
<b>Ejercicio 1.2</b>	<b><math>O(n)</math> en el algoritmo, <math>O(n)</math> para agregar y <math>O(n)</math> para buscar</b>	<b><math>O(n)</math> en el algoritmo <math>O(1)</math> para agregar y <math>O(n)</math> para buscar</b>
<b>Ejercicio 1.3</b>	<b><math>O(n^2)</math> para el algoritmo, <math>O(n)</math> para buscar</b>	<b><math>O(n^2)</math> para el algoritmo, <math>O(n)</math> para buscar</b>
<b>Ejercicio 2.1</b>	<b><math>O(n)</math> en el algoritmo, <math>O(n)</math> para agregar datos</b>	<b><math>O(n)</math> en el algoritmo, <math>O(1)</math> para agregar datos</b>

En la mayoría de los casos resulta más eficaz usar una lista enlazada debido a que su inserción de datos es muy rápida la hace en un tiempo constante, mientras que el vector dinámico se demora en un tiempo lineal  $O(n)$ , pero si estamos hablando de una búsqueda de datos resulta igual ya que ambas estructuras se demoran  $O(n)$  para buscar un elemento, la estructura de datos más eficiente para hacer una búsqueda y que tenga buena eficiencia a la hora de insertar elementos es un árbol binario debido a que hace la mayoría de las cosas en tiempo logarítmico, pero aún no sabemos cómo hacerlos.

**3.3 La complejidad de la clase BrokenKeyBoard es  $O(n)$  debido a que empleamos un solo for con unas operaciones dentro del que se realizan en tiempo lineal  $O(1)$ .**

**3.4 N es la cantidad de caracteres de la cadena de prueba.**

**PhD. Mauricio Toro Bermúdez**  
Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473



ESTRUCTURA DE DATOS 1  
Código ST0245

**4) Simulacro de Parcial**

4.1 B (OPC)  
4.2 A (OPC)  
2 C (OPC)  
3 PENDIENTE  
4.1 stack.pop() (OPC)  
4.2 C (OPC)  
5 A  
6 B  
7 FALTA AAAA  
8 C  
9.1 A  
9.2 C  
9.3 C  
10.1D  
10.2A  
10.3B  
11.1B  
11.2B  
12.1D  
12.2A  
13.1A  
13.2A  
14C

**PhD. Mauricio Toro Bermúdez**

Docente | Escuela de Ingeniería | Informática y Sistemas  
Correo: mtorobe@eafit.edu.co | Oficina: Bloque 19 – 627  
Tel: (+57) (4) 261 95 00 Ext. 9473

