

## 2º RECUPERATORIO DE ALGORITMICA Y ESTRUCTURA DE DATOS II - 2018

1) Incluir en el TAD **List** el método *RetainAll* y realizar su implementación en la clase **ArrayList**. Realizar una aplicación que pruebe el mismo.

```
/*
 * Retiene solo los elementos de la lista que están en la lista especificada.
 * En otras palabras remueve de la lista todos los elementos que no están
 * contenidos en la lista especificada.
 * Retorna true si la lista cambia como resultado de su llamada.
 */
public boolean retainAll(List<E> l)
```

2) Implementar un método en la clase **LinkedBinaryTree** que verifica si un árbol tiene elementos duplicados. Realizar una aplicación que pruebe el mismo.

```
public boolean duplicado()
```

3) Desarrollar una aplicación que cuente las palabras reservadas de java de un archivo. Cargan al menos diez palabras reservadas usando un árbol binario ordenado. Leer el archivo a procesar desde la línea de comandos. Al finalizar mostrar la ocurrencia de cada palabra reservada.

```
// Ejemplo para abrir un archivo y leer palabras
File file = new File(args[0]);
Scanner doc = new Scanner(file).useDelimiter("[^a-zA-Z]+");
while (doc.hasNext())
    System.out.println(doc.next());
```

---

### TEORIA (Entregar en una hoja aparte)

1. En el análisis de algoritmos, qué significa el estudio experimental y que deficiencias presenta. ¿Cuál es la propuesta superadora a este tipo de análisis?
2. ¿Qué es aporta una interfaz de Java al Tipo Abstracto de Datos (TAD)? Explique qué significa la implementación de un TAD en lenguaje JAVA.
3. ¿Qué es un iterador; y cuál es la ventaja que una implementación de TAD lo defina?
4. ¿Qué casos se deben tener en cuenta al programar el método de eliminación del nodo de un árbol binario de búsqueda?